AD-A062 845 DIALOG SYSTEMS INC BELMONT MA F/G 17/2

INTELLIGENCE REPORT VOICE INPUT.(U)

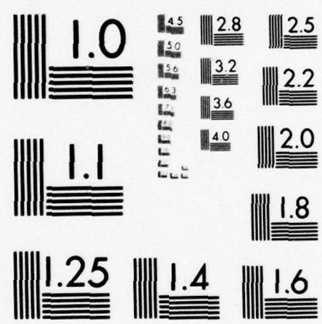
NOV 78 L BAHLER, P MARKEY, S MOSHIER F30602-76-C-0257

UNCLASSIFIED RADC-TR-78-239 NL

1 OF 1

AD
A062845

END
DATE
FILMED
2-79
DDC



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU

AD A062845

LEVEL II

RADC-TR-78-239
Final Technical Report
November 1978

INTELLIGENCE REPORT VOICE INPUT

Dialog Systems Incorporated

L. Bahler
P. Markey
S. Moshier

Approved for public release; distribution unlimited

ROME AIR DEVELOPMENT CENTER
Air Force Systems Command
Griffiss Air Force Base, New York 13441

79 01 02 004

This report has been reviewed by the RADC Information Office (OI) and is releasable to the National Technical Information Service (NTIS). At NTIS it will be releasable to the general public, including foreign nations.

RADC-TR-78-239 has been reviewed and is approved for publication.

APPROVED: Melvin G. Manor, Jr.

MELVIN G. MANOR, JR.
Project Engineer

APPROVED: Howard Davis

HOWARD DAVIS
Technical Director
Intelligence & Reconnaissance Division

FOR THE COMMANDER: John P. Huss

JOHN P. HUSS
Acting Chief, Plans Office

If your address has changed or if you wish to be removed from the RADC mailing list, or if the addressee is no longer employed by your organization, please notify RADC (IRAA), Griffiss AFB NY 13441. This will assist us in maintaining a current mailing list.

Do not return this copy. Retain or destroy.

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

19 REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER 18 RADC-TR-78-239	2. GOVT ACCESSION NO.
3. RECIPIENT'S CATALOG NUMBER	
4. TITLE (and Subtitle) 6 INTELLIGENCE REPORT VOICE INPUT.	5. TYPE OF REPORT & PERIOD COVERED 9 Final Technical Report,
	6. PERFORMING ORG. REPORT NUMBER N/A
7. AUTHOR(s) 10 L. Bahler, P. Markey S. Moshier	8. CONTRACT OR GRANT NUMBER(s) 15 F30602-76-C-0257
9. PERFORMING ORGANIZATION NAME AND ADDRESS Dialog Systems, Inc. 32 Locust Street Belmont MA 02178	10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS 62702F 16 45940623 17 06
11. CONTROLLING OFFICE NAME AND ADDRESS Rome Air Development Center (IRAA) Griffiss AFB NY 13441	12. REPORT DATE 11 November 1978
	13. NUMBER OF PAGES
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office) Same 12 89p.	15. SECURITY CLASS. (of this report) UNCLASSIFIED
	15a. DECLASSIFICATION/DOWNGRADING SCHEDULE N/A

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

Same

18. SUPPLEMENTARY NOTES

RADC Project Engineer: Melvin G. Manor, Jr. (IRAA)

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Automatic Word Recognition
Speech Recognition
Pattern Recognition

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

This work was an initial effort in the use of voice data entry for information data handling. The objective of this effort was to develop the technology for a large vocabulary (1000 word) isolated word recognition system capable of quick adaptation and high accuracy for a limited number of people. Techniques for word boundary detection, noise suppression, and frequency sealing were examined. Tests were conducted on a 1000 word and a 100 word unstructured vocabulary. Recognition accuracies of 30.5 ~~percent~~ and 66 ~~percent~~ were obtained for the untrained case and 62.4 ~~percent~~ and 90 ~~percent~~ after training each word once.

DD FORM 1 JAN 73 1473

% UNCLASSIFIED

392051

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

79 01 02 004

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE(*When Data Entered*)

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE(*When Data Entered*)

Table of Contents

List of Figures

List of Tables

EVALUATION

This contract was in support of TPO R1B, and has application to the voice input of information to data bases. The objective of this effort was to develop the technology for a large vocabulary (1000 word) isolated word recognition system capable of quick adaptation and high accuracy for a limited number of people. Reasonable accuracies were obtained for an unstructured large vocabulary and highlighted areas where additional technology development is required.

Melvin G. Manor, Jr.
MELVIN G. MANOR, JR.
Project Engineer

1.0 Summary

An isolated word speech recognition facility was used to test the recognition accuracy of a 1000 word speaker independent data base. Software and hardware improvements were designed, tested and implemented to accomodate the larger vocabulary. Two major sections of effort are discussed. The first section describes the speech recognition facility. This discussion includes the data collection process, and the hardware and software facilities which were developed during the contract period. The second section, Experiments, contains a detailed description of efforts to improve the recognition algorithm. Techniques for word boundary detection, noise suppression, and frequency scaling are examined with respect to their effects on recognition accuracy. Several strategies were employed to develop archetypical reference patterns and a detailed analysis of the results is included. Experiments on large vocabularies (1000 words) were conducted under several conditions using six speakers not included in the data base. These tests indicated that more speakers should be added to the data base and that the isolated word algorithm should be examined with a view of increasing recognition accuracy for multi-syllabic words. A general facility for investigating the effect of syntactic constraints on large vocabulary data bases was also developed. This facility was demonstrated to Government representatives and the results of these tests are included in the final section of the report.

2.0 Speech Recognition Facility

2.1 Data Collection

Isolated word speech samples were collected on location and in the laboratory from locally available individuals, members of various community groups, university students and faculty, and business people. For on location recording, a Teac cassette recorder (TRQ-2040D) with a Teledyne MC057 dynamic microphone was used. Laboratory recordings were made on a Teac 2300SX reel to reel tape deck with an AKG K108 headset microphone. Recording environments were, for the most part, quiet but not noise free. Field recordings of word lists as spoken by about 200 individuals were obtained (See Appendix A, 1000 Word List). Due to the length of time re-

quired to record all of the 1400 target words, in most instances, only partial lists were collected from each individual. About 130,000 samples were collected on the cassette recorder. These were copied onto 7 inch reel to reel audio tapes for further processing.

In our laboratory, two sets of recordings of each of 1430 words were obtained from each of six subjects. Automatic prompting, driven by a PDP-11/03 computer was employed to insure good isolation between words. The word list was randomized and presented on a CRT computer terminal at a rate of one word every 3 seconds. Each subject sat for 24 sessions each lasting about 20 minutes. Each of the resulting tape recordings was then transcribed by hand and preliminary problems such as mispronunciations and loud background noise were identified. These transcriptions were used as a guide during further processing.

2.2 Hardware

The audio signal was band limited by a high pass, 18 dB per octave 300 Hz filter and by a low pass sharp cutoff 3.6 KHz filter, and digitized at 8 KHz with 12 bits resolution. Two PDP-11/04 controlled analog to digital conversion systems were built to implement the data base gathering part of the task. Each system consisted of the following devices:

1. A Digital Equipment Corporation PDP-11/04 with 28 K words of memory.
2. A Dynastor Inc floppy disk drive and interface.
3. A Lear Siegler Inc ADM3 CRT and interface.
4. A Digi-Data 1600 BPI Phase encoded 9-track digital tape drive and interface.
5. A Tektronix XY oscilloscope display and interface.
6. A Teac 2300 SX 2-track audio tape recorder.
7. A Dialog Systems analog to digital and digital to analog interface.
8. A Computer Operations Inc link tape drive and interface.
9. A BGW Model-100 stereo audio amplifier and 2 Acoustic Research 17 speakers.

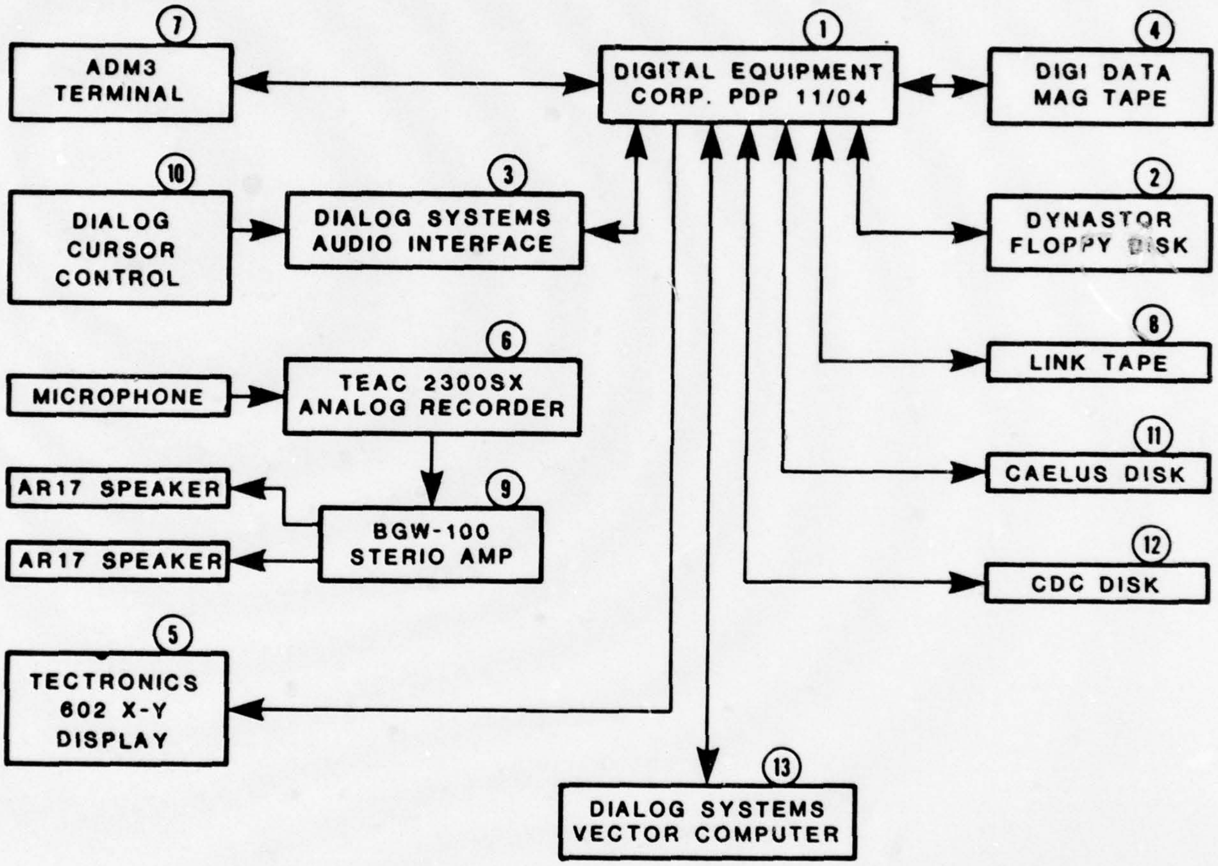


FIGURE 1 SPEECH PROCESSING FACILITY

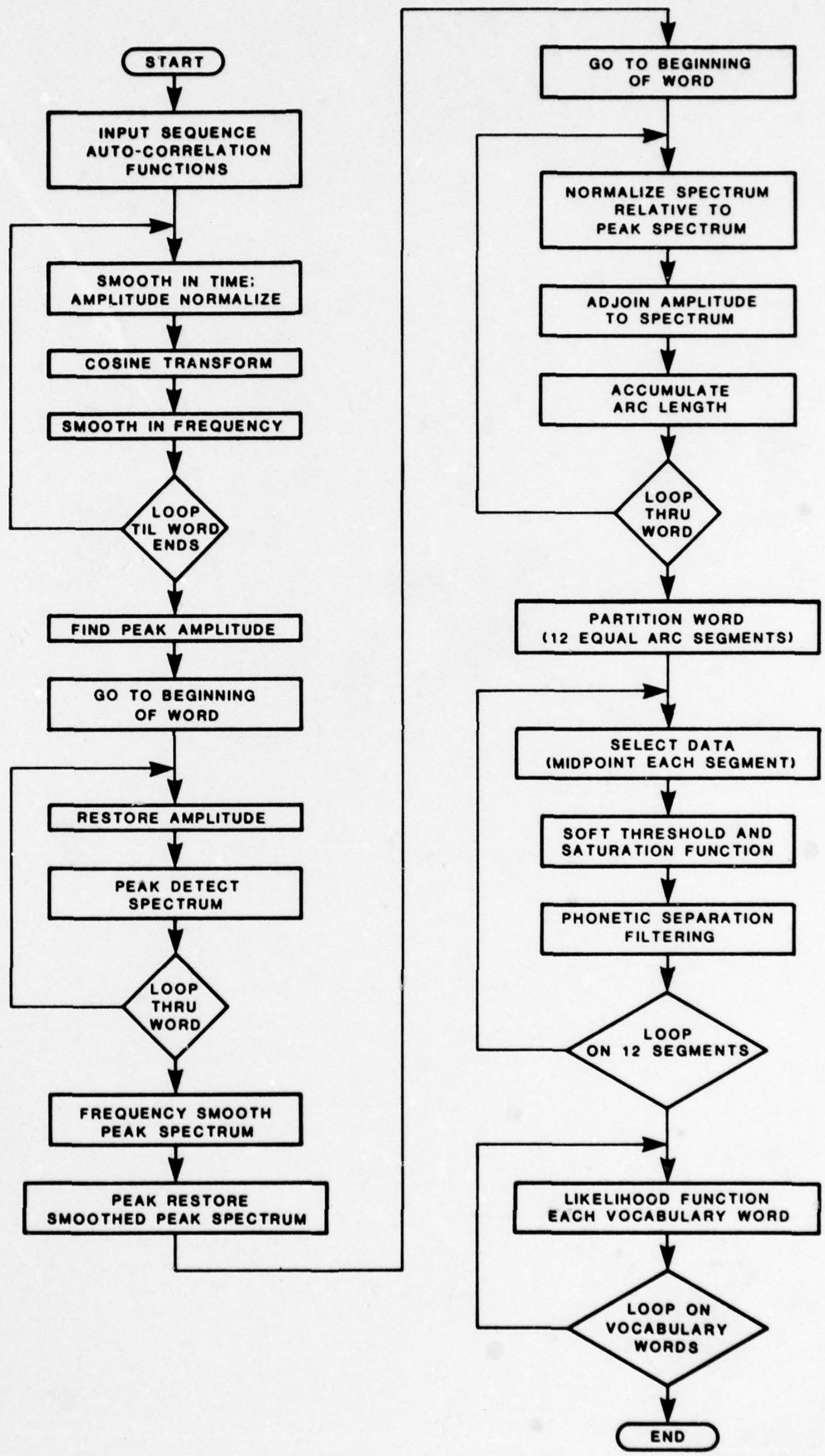


FIGURE 1A VOICE RECOGNITION PROGRAM FLOW DIAGRAM.

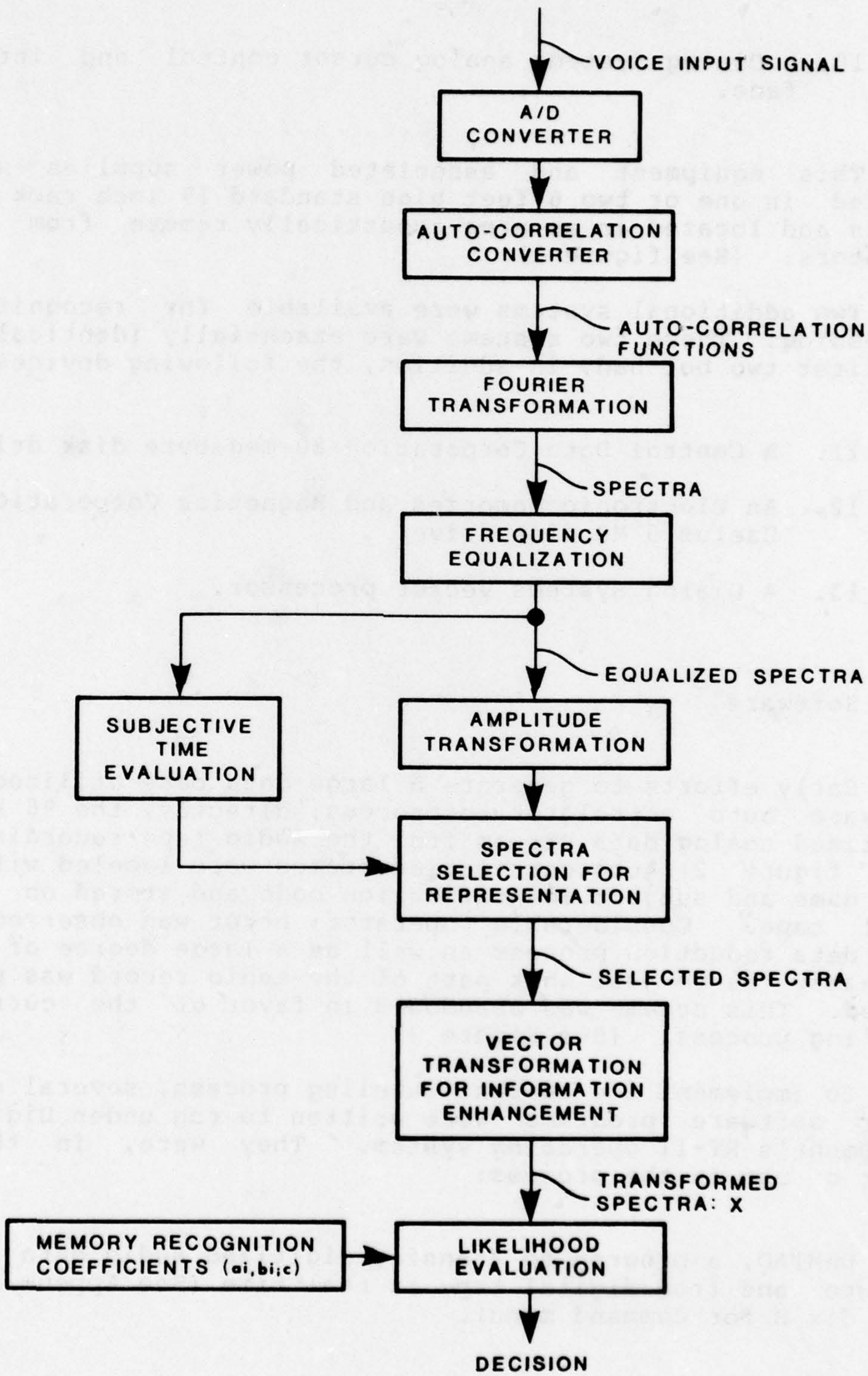

VOICE INPUT SIGNAL
A/D CONVERTER
AUTO-CORRELATION CONVERTER
AUTO-CORRELATION FUNCTIONS
FOURIER TRANSFORMATION
SPECTRA
FREQUENCY EQUALIZATION
EQUALIZED SPECTRA
SUBJECTIVE TIME EVALUATION
AMPLITUDE TRANSFORMATION
SPECTRA SELECTION FOR REPRESENTATION
SELECTED SPECTRA
VECTOR TRANSFORMATION FOR INFORMATION ENHANCEMENT
TRANSFORMED SPECTRA: X
MEMORY RECOGNITION COEFFICIENTS (ai,bi;c)
LIKELIHOOD EVALUATION
DECISION

FIGURE 2

10. A Dialog Systems analog cursor control and interface.

This equipment and associated power supplies were mounted in one or two 6 feet high standard 19 inch rack cabinets and located in an area acoustically remote from the operators. (See figure 1)

Two additional systems were available for recognition processing. These two systems were essentially identical to the first two but had, in addition, the following devices:

11. A Control Data Corporation 80-megabyte disk drive.

12. An Electronic Memories and Magnetics Corporation Caelus 5 MB disk drive.

13. A Dialog Systems vector processor.

2.3 Software

Early efforts to generate a large data base utilized a hardware auto correlator to process, directly, the 96 kbps digitized analog data stream from the audio tape recordings. (See figure 2) Auto correlation frames were labeled with a word name and subject identification code and stored on digital tape. Considerable operator error was observed in this data reduction process as well as a large degree of uncertainty as to just what part of the audio record was processed. This scheme was abandoned in favor of the current labeling process. (See Figure 3)

To implement the current labeling process, several computer software programs were written to run under Digital Equipment's RT-11 operating system. They were, in their order of use in the process:

DAMTAD, a program to transfer digitized audio data to and from digital tape in real time (See Appendix B for command menu).

VEDIT, a program to read digital tapes containing audio data, display audio wave forms, play back a corresponding acoustic wave form and generate a

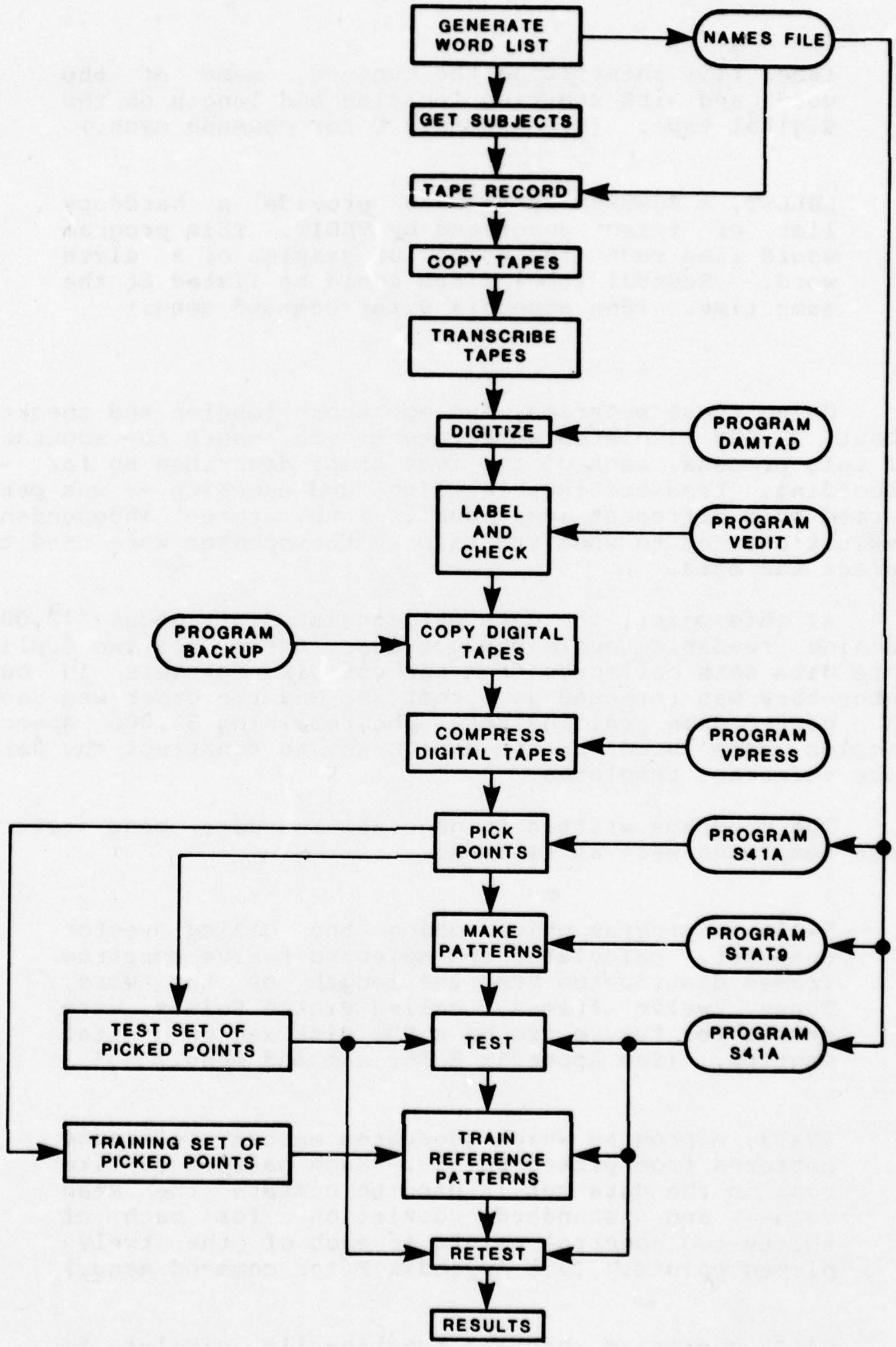


FIGURE 3 DATA BASE GENERATION

label file identifying the subject, name of the word and its starting location and length on the digital tape. (See Appendix C for command menu.)

LBLLST, a FORTRAN program to provide a hardcopy list of files generated by VEDIT. This program would also count the number of samples of a given word. Several label files could be listed at the same time. (See appendix D for command menu.)

Using these programs, two operators labeled and checked about 72,000 isolated word tokens. To insure the accuracy of this process, each of the four steps described so far -- recording, transcribing, labeling, and checking -- was performed by a different individual. Thus, three independent evaluations as to what was said by the speaker were used to select the data.

At this point, the data set consisted of about 72,000 machine readable audio productions. One of the two duplicate data sets collected from each of six speakers in our laboratory was reserved as a test set and the other was used as a post-design training set. The remaining 56,000 speech samples were used as the design set to construct the data base reference templates.

The programs written to generate the data base reference templates were as follows:

S4I36, a program which, using the Dialog vector computer, calculated and selected twelve spectrum frames distributed over the length of the word. These twelve frames, called Picked Points, were stored for future use on a CDC disk and on digital magtape. (See Appendix E for command menu.)

STAT9, a program which generates master reference patterns from picked points. Each pattern of like type in the data set is used to compute the mean value and standard deviation for each of thirty-two spectral points in each of the twelve picked points. (See Appendix F for command menu.)

S4IT, a program which is functionally similar to S4I36 but with the addition of modules which enable syntax tree manipulation. This program accepts real time audio input, picked point files,

or digitized audio files as input, and performs maximum likelihood recognition using reference statistics generated by STAT9. (See Appendix G for command menu and user interaction.)

3.0 Experiments

3.1 Algorithm Improvements

We have attempted to reduce erroneous samples that occur before and after the actual spoken word by measuring the background noise level each time the talker is cued to say a word. The amplitude threshold derived from the background level measurement has produced some improvement but has not eliminated these errors. If the recognition program should pick a frame containing voice only, the remaining chosen frames would, likely, be chosen in inappropriate places in the word. This type of error would generate reference templates with poor alignment and thus increase the correlation between adjacent picked points. We preserve the spectrum of the background noise and subtract this spectrum from each subsequent spectrm frame taken. This produces esentially zero movement of the subjective time function during intervals of background noise, and thus makes it nearly impossible for the algorithm to choose samples in these intervals. Under some conditions this approach may also improve the signal-to-noise ratio tolerance of the system.

This method operates by measuring the power spectrum of the input 50 milliseconds after the user is cued to say a word, and subtracting this power spectrum from the spectrum in each subsequent frame to the detected end of word. Any negative terms in the difference spectra are replaced by zeros. The operational performance of the algorithm was improved qualitatively, although there was only a slight improvement evidenced by tests of the digital data base (which excludes most data having noticeable background or cross talk noise). The noise subtracter did reduce the incidence of sample frames drawn erroneously from background noise, and this seems to be the source of a slight improvement in measured recognition accuracy.

To test the possibility that performance is limited by arithmetic truncation error in the likelihood calculation,

the software was modified to permit carrying extra precision in the reference patterns. There was a slight overall performance increase when the precision was increased from 8 to 9 bits, and no further increase from 9 to 10 bits. This result agrees with a similar finding several years ago with an earlier version of the algorithm. The current 8-bit precision will be retained.

The procedure by which representative samples are selected from an unknown word has been improved, but it appears that erroneous sampling decisions are still the largest source of error in the algorithm. To counteract this problem and allow for variant pronunciations, we have examined sample frequency histograms for some words and partitioned the data base to make two alternate reference patterns for these words. This partitioning has definitely improved the performance. For example, in a digit vocabulary with six of the words partitioned into alternate patterns, a preliminary test of a population of 110 male and female voices resulted in a correct recognition score of 99%. The alternate patterns should also improve the accuracy when the system is trained to the talker, since the sample variances represented in the reference patterns are more representative of particular pronunciations after the partitioning. No partitioning has yet been implemented in the 1000 word vocabulary experiments.

"Unbiased" likelihood parameters were implemented for isolated word training sets of 50 and 80 subjects. In each case the performance on the training set was closer to the performance on a large test set than when the "biased" parameters were employed. However, the improvements in prediction were not large enough to be considered statistically significant. A possible explanation lies in the fact that the variances of the individual pattern elements are in the same general range for all vocabulary words. Thus it is possible to reorder the pattern vectors so that corresponding elements of different word patterns have nearly the same variance. This makes the "biased" and "unbiased" likelihood ratio tests give essentially the same results, since a difference can arise only if the variances or the sample sizes are different. There continues to be an unexplained, statistically significant difference between the performance on training and test sets, even for training populations as large as 100 voices.

To match the behavior of the human auditory system more closely, we modified the spectrum analysis by changing the frequency scale to correspond with the S. S. Stevens "mel" scale. The pitch p and frequency f (Hz) in this scale are related by a power law which we took to be

$$p = \left(\frac{f-30}{1000}\right)^{0.65} \qquad (3.1)$$

The spectrum analysis was adjusted to give samples at equally spaced pitches between 300 Hz and 3400 Hz. There was no detectable change in recognition accuracy. However, because the system is insensitive to signals below 300 Hz or above 3400 Hz, it is likely that under operational conditions spurious signals such as power line noise and signals above the 3400 Hz cutoff of the aliasing filter would be suppressed, leading to better performance in the field.

One motivation for testing the pitch scale was to permit a "tuning" adjustment for different talkers based on translation in pitch. An experiment done some years ago involved translation in log frequency; the results were negative and the frequency scale seemed inappropriate, since the first formant and higher formants appeared to vary differently. The following tests were run, using the pitch scale:

1. The pitch of each frame of a word was translated so as to maximize the likelihood score for the correct-choice reference pattern. Then the scores for all reference patterns were computed and the recognition rate tabulated. In many instances translation bettered the score for the correct choice. But on the average the scores for wrong choices improved even more, producing a net performance deterioration.

2. After maximizing the score for the correct pattern as above, the translated data were saved to form a new data base from which a new set of reference patterns was computed. This procedure was iterated several times, always with the same result as above.

3. To "tune" with unknown data the procedure was modified by computing and examining all likelihood scores for the vocabulary, and finding the pitch translation which produced the best score for any of the reference patterns. Again, the overall performance worsened.

A possible problem with these experiments was that the initial reference patterns represented a large population, rather than a single talker, and perhaps the tuning produced divergent motions toward irrelevant features of the average patterns. A set of patterns for a typical single talker

could be used instead to initialize the procedure.

There remain also the questions whether the pitch scale is really appropriate, and whether translation on any scale should be the best method of tuning.

An arbitrary smooth mapping of the frequency scale can be well approximated by multiplying the spectrum vector by a suitable matrix. For example, translation is prodced by an off-diagonal matrix, and various nonlinear maps can be constructed with the aid of Lagrange interpolation polynomials. For reference data y and other data x the matrix T which minimizes the expected magnitude of y-Tx has a closed-form solution involving the correlation matrices R_{xx} and R_{yx}:

$$T = R_{yx} R_{xx}^{-1} . \tag{3.2}$$

Evaluation and testing of this "tuning matrix" T for many subjects would tend to show whether any talker dependent constant frequency mapping can improve recognition scores.

The table below gives the performance of the "official" isolated word algorithm and the "modified" algorithm (as described below) on "digits," "non-digits," and both. The "digits" consist of 3436 spoken words roughly evenly distributed over the ten digits, plus "yes" and "no," from both male and female speakers. The non-digits consist of 3940

Table 1

Recognition Rates (Old/New)

	Test Set		
Pattern	Digits	Non-Digits	Both
Digits	88.2/89.3	---	---
Non-digit	---	86.5/88.9	---
Both	81.6/82.0	80.5/84.9	81.0/83.6

spoken words roughly evenly distributed over the words:

affirmative, negative, close, file, briefing, amend, niner, specialist, north, south, east, west, and local, from both male and female speakers. No alternate patterns were used. The recognition rates were found by scoring the reference patterns on the same data base that was used in generating them. The recognition rates for the "official" algorithm were adjusted to include "duration" errors ("too-soons," timeouts, etc.) that were not found in the "modified" algorithm.

Reported below are changes made to the isolated word algorithm and some of their measured effects on recognition rate.

The following changes have been made to a version of S4I:

I. Noise subtractor:

A. Any sub-threshold frames are zeroed.
B. The noise-defining frames (1-6) are also zeroed.

II. Beginning-of-word detector:

A. To find the beginning of word:

1. Find the first frame where NTHRSH out of BWIDTH frames are above threshold.
2. Back up NBKUP frames from II-A-1.
3. First frame at or to right of II-A-2 that is above threshold is defined as beginning of word.

#NT sets NTHRSH (default 6)

#BW sets BWIDTH (default 12)

#NB sets NBKUP (default 8)

Table 2 gives the effect of BWIDTH on recognition rate. Table 3 gives the effect of NBKUP and NTHRSH on recognition rate. The values of these parameters do not appear to be too critical, with the values used initially (defaults) working as well as any.

B. All frames before the onset of listening are considered silent frames (zeroed). (This together with I-B makes the treadmill unnecessary.)

III. End-of-word detector:

A. To find the end of word:

1. Find first frame after beginning of word where EWIDTH contiguous frames are sub-threshold.

2. Run beginning-of-word detector backwards in time from III-A-1.

#EW sets EWIDTH (default 24)

Table 4 gives the effect of EWIDTH on the recognition rate. Again, the value of this parameter does not appear very critical.

B. All frames after the end of the input stream (i.e. after the end label for an A/D file input) are considered silent frames (zeroed). Overflowing the JIN buffer is now an error.

IV. Linear Time Point Picking:

The twelve points are picked, equally spaced, in real time from the beginning of word (first point) to the end of the word (last point).

V. Averaged Noise Sample:

The noise sample used by the noise subtractor consists of the average of the third and the sixth frames. The threshold used by the beginning-of-word detector consists of twice the average of the amplitude of the third through the sixth frames.

Using an averaged noise sample increased the recognition rate by 0.9%.

TABLES 2-4

BWIDTH = Width of beginning-of-word characteristic function

NTHRSH = Number of frames needed above threshold for beginning of word

NBKUP = Number of frames to back up at beginning of word

EWIDTH = Number of contiguous sub-threshold frames needed at end of word before applying symmetrical end-of-word detector

Table 2

Effect of BWIDTH on Recognition Rate

BWIDTH	Recognition Rate
6	88.9
9	88.9
12	89.2
15	89.1

NTHRSH = 6

NBKUP = 8

EWIDTH = 18

Table 3

Effect of NBKUP and NTHRSH on Recognition Rate

	NBKUP						
NTHRSH	2	4	6	8	10	12	14
3	88.7	89.1	89.3	88.9	88.9		
6	87.1	88.4	89.1	89.2	88.9	89.0	
9		85.3	87.8	88.6	88.6	89.0	89.1

Recognition Rates

BWIDTH = 12 EWIDTH = 18

Table 4

Effect of EWIDTH on Recognition Rate

EWIDTH	Recognition Rate
12	89.0
15	89.1
18	89.2
18	89.2
21	89.2
24	89.3
27	89.3

NTHRSH = 6

NBKUP = 8

BWIDTH = 12

3.2 Archetypical Reference Patterns

Software to determine a set of single spectrum frame archetypes was developed. The initial strategy was to partition the amplitude range at each freqency into p equal segments. With 32 freqencies in each spectrum there would thus be p possible cells. A data base of about 1500 words (18,000 frames) was analyzed by determining which cell each frame fell into and counting the number of frames in each cell. For p>2 nearly every cell had less than three occupants, making it impossible to tell where the natural clusters were located. With p=2, and considering only the lowest 16 frequencies, the occupancy distribution of Figure 4 was observed.

A set of the most densely occupied cells was chosen, and a reference pattern for each cell made by computing means and standard deviations of the 32 spectrum points for all the frames contained in the cell. The resulting mean spectra exhibited characteristic formant resonances but did not in general resemble the reference patterns derived from labeled words. Apparently only the speech frames showing relatively narrow (high-Q) resonances or essentially no resonance had sufficiently consistent behavior near the average spectral amplitude to show up as clusters, and the detection of cluster centers was biased by this effect. The standard deviations were essentially constant with frequency in the region defined by the Euclidean cells, reflecting

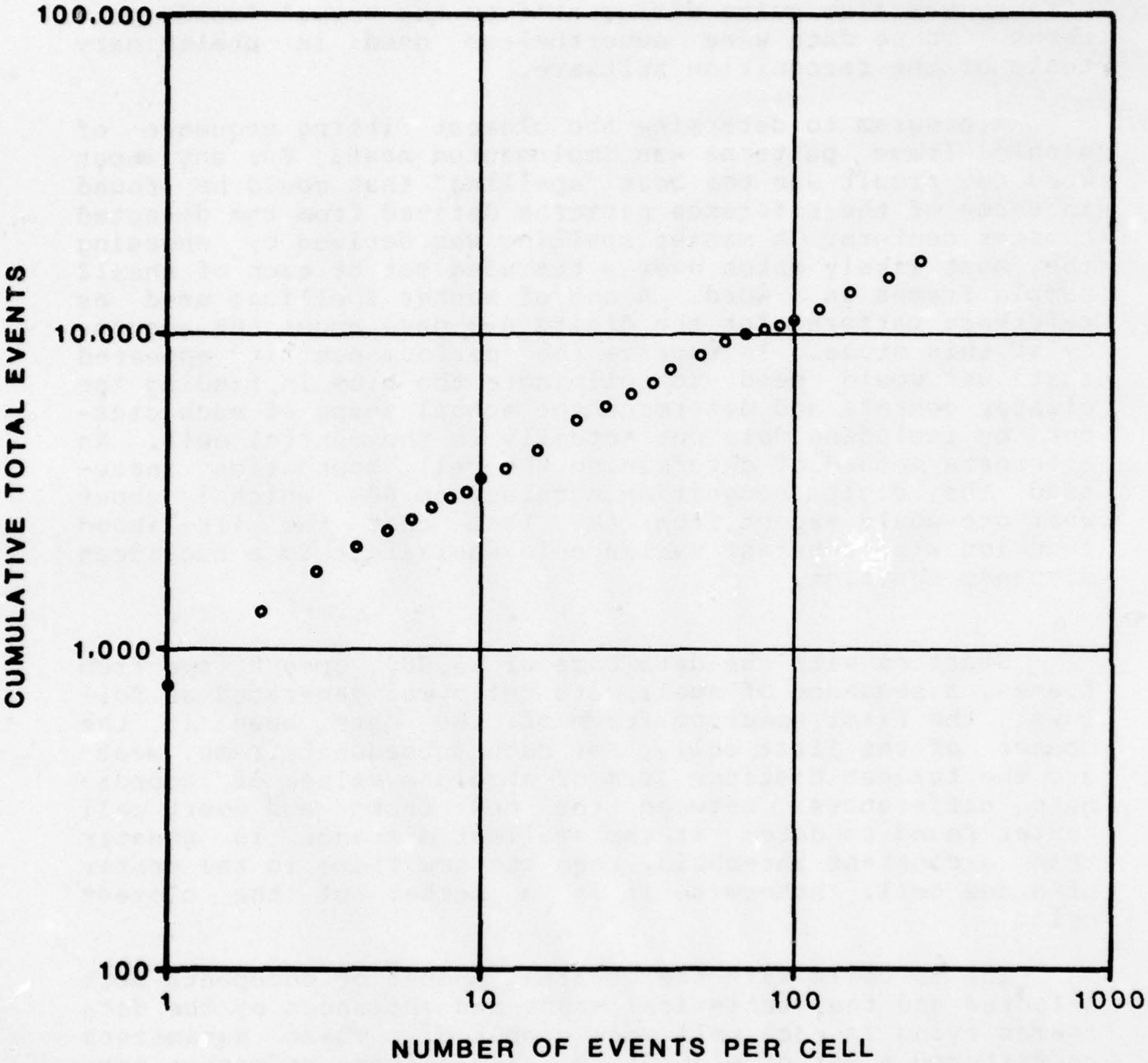


FIGURE 4 CUMULATIVE DISTRIBUTION FUNCTION OF SPEECH EVENTS RANKED BY POPULATION OF ANALYSIS CELLS.

perhaps a uniform distribution of data in each cell. This effect was also quite different from the normal "word" patterns. These data were nevertheless used in preliminary tests of the recognition software.

A program to determine the closest fitting sequence of single frame patterns was implemented next. For any input word the result was the best "spelling" that could be found in terms of the reference patterns derived from the detected cluster centers. A master spelling was derived by choosing the most likely match over a training set at each of the 12 sample frames in a word. A set of master spellings used as reference patterns for the digits 0-9 gave about 60% accuracy at this stage. To improve the performance it appeared that we would need to eliminate the bias in finding the cluster centers and determine the actual shape of each cluster by including data not actually in the central cell. An alternate method of determining the cell boundaries increased the digit recognition accuracy to 80%, which is about what one would expect from the fact that the likelihood function with constant variance is equivalent to a Euclidean distance function.

Starting with the data base of 18,000 speech spectrum frames, a sequence of small data cells was generated as follows: the first spectrum frame of the data base is the center of the first cell. For each subsequent frame, measure the taxicab distance (sum of absolute values of coordinate differences) between the new frame and every cell center found to date; if the smallest distance is greater than a constant threshold, then the new frame is the center of a new cell; otherwise it is a member of the closest cell.

The 65 cells with the greatest number of occupants were selected and the statistical means and variances of the data frames lying in each cell were computed. These parameters constituted a set of elementary, single-frame reference patterns. The data base of 1500 isolated words was then analyzed by computing the likelihood statistic for each frame of each word with respect to each of the 65 elementary patterns. At each word frame the six highest scoring pattern matches were tablated. A master spelling for each word was then derived by selecting at each frame the most frequently-appearing pattern among the tabulated ones.

A recognition test was then conducted by summing the pattern likelihood scores over each master spelling and choosing the spelling with the best cumulative score as the final word decision. Spoken digits tested in this way had an overall talker-independent recognition accuracy of 81%, which corresponds to results for a Euclidean distance function with one reference pattern per word.

Next, the 65 elementary reference patterns were refined by including all frames of the data base that were not already members of the 65 chosen cells. Each non-member frame was compared to the chosen cells by computing the likelihood statistic with respect to each of the 65 elementary reference patterns. Each frame was thereby assigned to the cell yielding the highest likelihood score. A new set of likelihood parameters was then computed for all the cells. With these parameters the recognition accuracy for digits was the same as before. Increasing the number of elementary patterns to 90 did not change the situation.

On the presumption that the above results were limited by particular choice of spellings, the following method of determining alternate word spellings was tested. Starting with the above set of "most popular" spellings for digits, the algorithm attempted to recognize words in a digit data base. At each error, the spelling (i.e. the sequence of elementary patterns having the best likelihood score at each frame of the word) for the mistaken word token was added to the list of alternates for that vocabulary word. After two training passes throuth a set of 150 digits (15 talkers) there were 3.2 spellings per digit on the average and zero errors. Testing on 20 additional talkers yielded 84% recognition. A test of "difficult" digits obtained under different conditions resulted in 55% to 65% recognition scores in the several tests described.

The principal conclusion to be drawn from these experiments is that the reference patterns found do not provide a particularly good estimate of the conditional probability densities of spoken digits.

The objective was to find unconditional densities (independent of any particular vocabulary word) which could be selected to build conditional densities for specific words. The variance terms of these reference patterns tended to be almost constant with frequency, even after including data frames not in the original cells. This tended to make the likelihood functions into Euclidean distance functions which could be made to yield reasonable recognition only by providing a number of alternate patterns for each vocabulary word.

Since the computation of likelihood functions is a great deal more time consuming than the process of accumulating word scores, experiments to examine the tradeoff between number of patterns and number of spellings were planned.

To examine the tradeoff between the number of elementary reference patterns and number of "spelling" templates a data base of digits spoken by 100 male and female talkers

was analyzed. For each test condition the number of templates required to obtain 99% recognition of this data base as a training set was tabulated. The reference patterns were derived by initializing clusters of frames having a constant taxicab metric radius followed by inclusion of all frames into a selected set of cells on the basis of likelihood ratio. With the number of selected cells as the variable, the following data were obtained:

Relative Radius of Initial Cell	Number of Elementary Patterns	Average Number of Templates per Word
1500	33	39
1024	60	35
1024	90	34
(conditional density)	120	27

Variations in conditions, such as a change in the radius of the initial cells, did not affect the results appreciably. The conclusion appears to be that this general method of deriving reference patterns requires a large number of spelling templates, and that the number of templates is not very sensitive to the number of elementary patterns.

To differentiate between the effects of particular pattern choices and the method of determining template spellings, a standard set of patterns was selected for a series of experiments on the method of choosing representative spellings. The patterns consisted of all the conditional density parameters obtained by partitioning the digit reference patterns used in the standard 12-frame reference patterns with one complete pattern per vocabulary word. It was reasoned that since these reference patterns yield the best available recognition scores, a good template-finding procedure should be able to discover the spellings corresponding to the sequence before the 12-frame patterns were partitioned. It turned out that the method of adding a spelling whenever an error occurred did not find the original reference pattern sequence. However, a modification of this technique was found that produced recognition accuracy as good as that of the standard algorithm.

The modified procedure was to permit the reference frame to be chosen independently for each frame comprising

the word pattern. Thus, the template for a vocabulary word consists of a set of alternate pattern names for each frame position in the word. Any spelling consisting of a sequence of one alternate elementary pattern chosen at each position in the word is a legal spelling. The likelihood score for the word is then the sum of the scores of the best-fitting pattern at each of the positions. The method is similar to the strategy used in our continuous speech keyword detection system.

For the selected data base of 1000 digits, the standard isolated word algorithm with one 12-frame reference pattern per word scored 91% correct recognition. The following table shows the results of the strategy just described as a function of the number of alternate elementary patterns established at each sample frame. Comparable results for the elementary patterns derived by the constant taxicab metric method are also shown. Reference patterns are elected by counting the number of times their likelihood scores appear in the top six choices for a given set of exemplar patterns.

Table 5

Percent Talker-Independent Recognition as a Function of Number of Alternate Patterns at Each Word Segment

Number of Alternates:	1	2	3	4	5	6	7
Conditional Density Parameters:	89	89	90	90	91	90	90
Unconditional Cluster Parameters:							
60 Patterns			83	84	83	84	84
90 Patterns			84			86	85
120 Patterns			84		86	86	86

The revised template finding procedure yielded significantly better recognition performance than the earlier method, and it can be seen from the table that performance did not change much as a function of the total number of elementary patterns.

An intermediate method of template finding was imple-

mented. In this method, consecutive pairs of elementary patterns constituted a sequence of sub-word spellings, and alternates were chosen freely at each sub-word position. The objective was to increase recognition accuracy at a given total number of elementary likelihood calculations by tightening the restriction on allowable template sequences at a cost of increasing the complexity of the template search.

For each test condition (number of alternate patterns permitted) the results were marginally better, but the improvement was 0.5% or less. At 5 alternate choices per comparison the elementary patterns consisting of either one frame or an ordered pair of comparison frames yield essentially the same performance as the standard set of 12-frame reference patterns. The results to date suggest that comparable performance cannot be obtained with unconditional density functions.

In smaller vocabularies for which large data bases are available, much effort has been expended trying to produce alternate statistical patterns to better represent the words. We experimented with an isolated word data base, consisting of 6197 exemplars of the ten digits and /yes/ and /no/. Using four alternates per word, statistics were compiled from the data base. The same data base when tested gave a 12-alternative forced-choice identification rate of 95% correct; the rate was 93% for an unknown test set of 900 words. The alternates were produced by manually labeling male, female, general and Southern dialects, followed by an iterative clustering algorithm.

We explored whether the number of alternative patterns used to represent each word was currently limiting the performance. If we attempted to increase the number of alternatives indefinitely, the alternatives would be highly trained to the exemplars over which their statistics were collected. Thus these exemplars would have to be excluded as possible matches, because of the training effect in our Gaussian maximum likelihood decision model. To circumvent these training effects, a nonparametric model was used, in which each exemplar in the data base was compared to each other word by computing their separation as measured by the Euclidean distances.

In the twelve alternative forced-choice experiment, in which each example was identified as the name of its nearest neighbor, performance was 92% correct. Each of the 6197 examples was compared to the 6196 other exemplars.

An experiment based on the maximum likelihood choice over a region was also run. For each unknown word, the set of 100 closest neighbors was found by using the Euclidean

distance measure. Then the unknown word was identified as the word occurring most often in that set of neighbors. Performance was the same.

To model the effect of using the most neighbors rather than the nearest neighbor as the decision rule, assume that, in a neighborhood about an unknown exemplar, there are uniformly distributed exemplars from two words, w_1 and w_2, with relative frequencies p_1 and p_2, respectively,

$$p_1 + p_2 = 1. \tag{3.3}$$

Also assume that these measured relative frequencies are the true probabilities for an unknown exemplar in this neighborhood. The maximum likelihood decision rule would identify the unknown as word w_i if $p_i > p_j$ for each j, to give the best expected correct identification rate, equal to max (p_1, p_2). This decision rule corresponds to identifying with the most neighbors. On the other hand, the nearest neighbor would be word w_i with probability p_i.

Thus averaged over the data base, the nearest neighbor rule picks w_i in proportion to its relative frequency p_i, giving a correct identification rate of

$$p_1^2 + p_2^2,$$

which is less than or equal to max (p_1, p_2), assuming p_1 and p_2 are less than or equal to 1.

To identify a word as its nearest neighbor is a sub-optimal

rule. Depending on the actual values of the relative frequencies, the maximum likelihood error rate E_{ML} is bounded between the closest neighbor error rate E_{cn} and one-half E_{cn}.

We conclude that the performance is not limited by the number of alternatives used in representing the statistics, but that performance is limited by the instrinsic overlap of words' patterns as produced by present preprocessing techniques.

3.3 Large Vocabulary Tests

Preliminary tests were conducted on portions of the data base as they became available. Initially, a statistics file was generated from approximately 1430 different lexical items as spoken by six individuals. In this experiment the test subject was included in the data base and represented one-sixth of the data base. When tested, 53% of 1404 lexical items were recognized. In this and all following experiments, each test word was compared to all words in the data base. All gross mispronunciations had been deleted from the test sets.

A design set of approximately forty speakers, male and female, consisting of approximately 53,000 speech samples was used to generate a data base. (See Appendix H, Audio Tape List) Two test conditions were examined. In the first condition, six speakers not included in the design set, were tested against the data base. The results, summarized below, for 6315 lexical events, produced an average recognition of 30.5%. Each lexical event was tested against each of 1078 lexical items in the vocabulary, where recognition was defined as a match between the most likely items and the name of the event. The vocabulary was not selected for good phonetic distribution; thus, for example, the spoken letter "b" and the word "be," the letter "c" and the word "sea," all are different vocabulary words.

The second condition tested the recognition accuracy against the data base after a single training trial with a weight of 50%. That is, the mean value of a spectral point

Table 6

1000 Word Recognition Rate, Untrained

Subject Number	Number of Events	Number of Errors	% Correct
0	1056	708	32
1	1054	818	22
2	1055	779	26
3	1055	697	33
4	1050	668	36
5	1045	700	33
Totals: 6	6315	4390	30.5

in the reference pattern was averaged with the value of a like spectral point from the training sample. A second, independent sample of each word as spoken by each subject from the first test condition was used to evaluate recognition accuracy after training. These data are summarized in tabular form below. In 6311 lexical events tested against the trained data base of 1078 lexical items, 4390 errors occurred. The resulting recognition accuracy was, averaged over six subjects, 62.4%.

Table 7

1000 Word Vocabulary Recognition Accuracy, Trained with One Sample of Each Word

Subject Number	Number of Events	Number of Errors	% Correct
0	1053	388	67
1	1051	405	61
2	1050	325	69
3	1055	395	62
4	1049	426	59
5	1053	481	54
Totals: 6	6311	2370	62.4

Additional experiments were run on the 1400 word vocabulary. The nearest Euclidean neighbor decision rule was used. There were two recordings of each of six subjects, numbered 0 through 5. One set of recordings was used as a reference set of up to 8413 patterns; the other was the test set. Results are summarized in the table below.

Test Set Subjects	Number of Trials	Reference Subjects	Number of Patterns	Percent Correct
0-3, part of 4	6563	0-5	8413	53.1
0	1404	0	1408	54.3
0	1404	1-5	7005	26.0

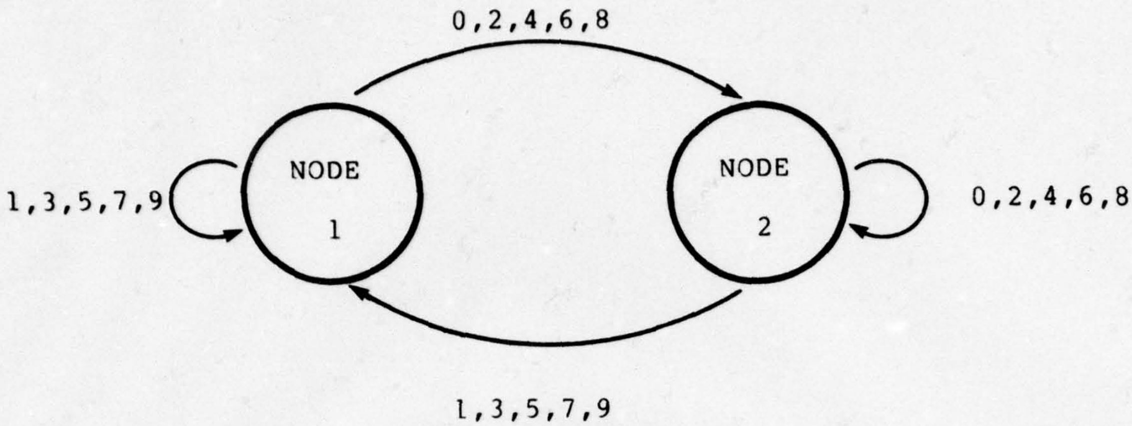
Because there are only six tokens per word it is hard to draw any conclusion about the intrinsic overlap of the word pattern distributions. Examination of actual pattern data does support the idea that many or most errors are due to gross misalignment of spectrum sampling times between unknown and reference word patterns.

3.4 The Syntax Tree Facility

In some applications, only words in a subset of the total stored vocabulary are legal alternatives at any given point. A facility exists for entering such constraints into a syntax tree and for traversing the tree, significantly improving recognition accuracy and response time for large vocabularies.

Each syntax tree consists of a set of numbered nodes and named branches between the nodes. The name of a branch corresponds to the vocabulary word that can cause a transition between the spanned nodes. For example, in the simple tree below, saying an odd digit will cause a transition to node 1 and saying an even digit will cause a transition to node 2:

Run-time commands exist for invoking a tree, moving around in a tree, and printing out information on a tree.



Furthermore, arbitrary subroutines may be associated with particular nodes so that a transition to a node can invoke code. The path taken through the tree is saved and transitions can be taken back under both keyboard and voice control.

Non digits from the 100 required word list (Appendix I) were entered at a node in the syntax tree and tested by a government representative who was not included in the data base. Of the 67 words in the node, 66% were recognized correctly. After training each word once, 90% were recognized.

APPENDIX A

1000 Word List

1500	A
1600	ABLE
0502	ABOVE
0400	ACCOUNT
0503	ACKNOWLEDGE
0504	ACKNOWLEDGED
0401	ACT
0372	ACTION
1400	ACTIVATE
0403	ACTIVITY
0384	ADD
0506	ADDED
0404	ADDRESS
0507	ADDRESSEE
0405	ADMINISTRATION
0510	AFRICA
0511	AFRICAN
0512	AFTER
0513	AGAIN
0514	AGAINST
0515	AGE
0406	AID
0409	AIM
0407	AIR
0408	AIRFORCE
1474	ALARM
0517	ALGERIA
0518	ALL
0410	ALLIANCE
1800	ALPHA
0519	ALREADY
0522	ALWAYS
0411	AMBASSADOR
0523	AMERICAN
0412	AMMUNITION
0524	AMOUNT
0526	AN
0414	ANALYSIS
0527	ANCIENT

1000 Word List

0528	AND
1465	ANSWER
0530	ANTI
0531	ANY
1442	APOSTROPHE
0534	APPREHEND
0536	APPROVE
0537	APPROVED
0539	APRIL
0541	ARAB
0542	ARABIAN
0543	ARCTIC
0544	ARE
0545	AREA
0546	ARGENTINA
0415	ARMEDFORCES
0416	ARMY
0549	AROUND
0550	ARREST
0417	ARTILLERY
0551	AS
0552	ASIA
0553	ASKED
0555	ASSOCIATE
0556	ASSOCIATED
1460	ASTERISK
0557	AT
0558	ATLANTIC
0418	ATOMIC
0420	ATTACHE
0559	ATTACK
0419	ATTENTION
0421	ATTRITION
0561	AUGUST
0562	AUSTRALIA
0563	AUSTRIA
0422	AUTO
0564	AUTOMATIC
0423	AVENUE
1501	B
0424	BACK
1443	BACKSPACE
0325	BACKWARD
0570	BAD
1601	BAKER
0571	BALL
0425	BANK
0426	BANKER
0427	BASE
0575	BASES
0429	BATTALION
0428	BATTLE
0577	BAY

1000 Word List

0578	BE
0579	BEACH
0582	BECAUSE
0583	BECOME
1464	BEDLIGHT
1414	BEDMOTOR
0585	BEEN
0586	BEFORE
0587	BEIRUT
0588	BELGIUM
0589	BELIEVE
0591	BENT
0593	BEST
0594	BETTER
0595	BETWEEN
0596	BIG
0597	BIOLOGICAL
0363	BLACK
0430	BLOC
0598	BLOODY
0367	BLUE
0431	BOAT
0599	BOILING
0432	BOMBERS
0433	BOMBS
0602	BOTH
0434	BOUNDARIES
1441	BRACKET
1801	BRAVO
0604	BRAZIL
0435	BRICK
0608	BRIEF
1412	BRIGHTER
0609	BRITAIN
0610	BRITISH
0611	BROOK
0436	BUG
0612	BUILD
0437	BULLETS
0615	BURMA
0616	BURN
0438	BUSINESS
0617	BUT
0618	BUYING
1448	BY
1502	C
1445	CALCULATOR
1440	CALL
0621	CALM
0623	CAME
0439	CAMPS
0624	CAN
0625	CANADA

0626	CANAL
0322	CANCEL
0628	CANNOT
1426	CAPITAL
0440	CARE
0441	CARGO
0442	CARRIER
0443	CASE
1602	CAST
0444	CASUALTY
0632	CATCH
0634	CAUGHT
0636	CENTER
0637	CENTURY
1403	CHANNEL
0638	CHANGE
0639	CHAOTIC
0445	CHARACTER
0640	CHARGE
1802	CHARLIE
0446	CHIEF
0645	CHINA
0646	CHINESE
0647	CHRISTIAN
0649	CIRCULAR
0650	CITY
0651	CIVIL
0447	CLARIFICATION
0356	CLEAR
0655	CLIFF
1459	CLOSE
0448	CLOTHES
0656	CLOUDS
0449	COALITION
0659	COLD
0660	COLLATE
1432	COLON
0455	COLONEL
0346	COLOR
0664	COME
1430	COMMA
0665	COMMAND
0450	COMMERCE
0666	COMMON
0668	COMMUNIST
0451	COMPANY
0700	COMPLETED
0701	COMPLEX
0704	CONCEAL
0706	CONCERNED
0371	CONDITION
0709	CONDUCT
0710	CONDUCTED

1000 Word List

0452	CONFRONTATION
0453	CONGRESS
0454	CONGRESSMAN
0457	CONSUL
0458	CONSULATE
0456	CONTEST
0717	CONTINENT
1471	CONTROL
0720	CONVENTIONAL
0459	CONVERSATION
0460	CONVICTION
0722	COOL
0724	COPY
0461	CORPS
0463	COSINE
0725	COULD
0727	COUNTRY
0462	COURSE
0728	COVER
0464	CRAFT
0730	CRETE
0731	CROOKED
0465	CRUISER
0733	CUBA
0734	CURE
0735	CURRENT
0328	CURSOR
0466	CURTAIN
0467	CUSTOMS
0737	CYLINDRICAL
0738	CYPRUS
1503	D
0740	DAMASCUS
1433	DASH
0741	DATE
0468	DATA
0742	DAY
0743	DEAD
0744	DEAR
0745	DECADE
0746	DECEMBER
0747	DEEP
0469	DEFENDERS
0470	DEFENSE
1444	DEFINE
1470	DELETE
0749	DELIVER
0750	DELTA
0751	DEMOCRATIC
0752	DEMOLISH
0753	DENMARK
0755	DEPLOY
0758	DEPRESSED

1000 Word List

0761	DESERT
0762	DESTROY
0471	DESTROYER
0763	DETERIORATING
0764	DEVELOP
0472	DEVELOPMENT
1419	DIAL
0473	DIALOGUE
0474	DICTATOR
0765	DID
0766	DIED
0475	DIESEL
1413	DIMMER
0767	DIRECT
0768	DIRECTED
0772	DISPATCHED
0774	DISPLAY
0777	DIVING
0476	DIVISION
0778	DO
0477	DOCK
0478	DOCTOR
1603	DOG
0479	DOLLAR
0352	DONE
0779	DONT
0329	DOWN
0781	DRY
1504	E
0783	EARLY
0784	EARTH
0053	EAST
1604	EASY
0785	EAT
0786	EBB
1804	ECHO
0787	ECONOMIC
1469	EDIT
0481	EFFORT
0788	EGYPT
0789	EGYPTIAN
0008	EIGHT
0308	EIGHTEEN
0316	EIGHTY
0790	EITHER
0791	ELECTRIC
0482	ELECTION
0793	ELECTRONIC
0795	ELEVATED
0333	ELITE
0483	EMBARGO
0484	EMBASSY
0485	EMBLEM

0797	EMPLACEMENT
0798	ENCLOSED
0799	END
0378	ENDTAPE
0802	ENGLAND
0486	ENROUTE
0351	ENTER
0803	ENTIRE
0805	ENTRY
0806	ENVIOUS
1401	ENVIRONMENT
0487	ENVY
0808	EQUAL
0355	EQUALS
0488	EQUIPMENT
0321	ERASE
0809	ESCAPE
0489	ESTIMATE
0811	ESTIMATED
0812	EUROPE
0490	EVACUATION
0813	EVALUATE
0814	EVEN
0815	EVENTUAL
0816	EVER
0817	EVERY
0818	EVOLVE
0819	EXCEED
0820	EXCEPT
0821	EXCITE
1438	EXCLAMATION
1421	EXECUTE
0491	EXERCISE
0822	EXPAND
0492	EXPANSION
0823	EXPECT
0824	EXPELL
0825	EXPLAIN
0826	EXPLORE
0827	EXPLOSIVE
0493	EXPLOSIVES
1456	EXPONENT
0831	EXTERIOR
0832	EXTERNAL
0833	EXTINGUISH
0834	EXTRA
0835	EXTRACT
0836	EXTREME
1505	F
0494	FACE
0837	FACILITY
0838	FACING
0496	FACT

0497	FACTION
0498	FACTOR
0840	FACTUAL
0841	FADE
0842	FAIR
0843	FAITHFUL
0844	FALL
0845	FALSE
0846	FAMILIAR
0847	FASCIST
0848	FAST
0849	FASTER
0850	FAT
0851	FATAL
0852	FATIGUE
0499	FAULT
0853	FAVOR
0854	FAVORITE
0855	FEAT
0856	FEBRUARY
0857	FEELER
0858	FEELING
0860	FELL
0863	FEUD
0864	FIELD
0865	FIERCE
0305	FIFTEEN
0866	FIFTH
0313	FIFTY
0867	FIGHTERS
0868	FIGHTING
0869	FIGURE
0013	FILE
0870	FINAL
0871	FIND
0873	FINLAND
0874	FIRE
0876	FIRST
0877	FISSION
0005	FIVE
0879	FIX
0880	FIXED
0881	FLAME
0882	FLAG
0883	FLAW
0884	FLED
0885	FLEE
0886	FLEW
0887	FLIGHT
0889	FLO D
0890	FLOTILLA
0891	FLOW
0892	FLY

0893	FOG
0894	FORCE
0895	FOREIGNERS
0896	FOREST
0897	FORM
0334	FORMAT
0898	FORMATION
0899	FORT
0900	FORTIFICATION
0901	FORTIFIED
0312	FORTY
0324	FORWARD
0902	FOUND
0004	FOUR
0304	FOURTEEN
0904	FOURTH
1605	FOX
1805	FOXTROT
0905	FRAME
0908	FREEZING
0910	FRENCH
0912	FRESH
0913	FRIGHT
0914	FROM
0915	FRONT
0916	FUEL
0917	FURTHER
0918	FUSION
0919	FUTURE
1506	G
0920	GALE
1468	GAME
0923	GENERAL
0924	GENERATE
1606	GEORGE
0926	GERMANY
0927	GET
0929	GIVE
0930	GLAD
0931	GLADLY
0932	GO
0933	GOING
1806	GOLF
0934	GOOD
0935	GOT
0335	GRAPHICS
0939	GREAT
0940	GREECE
0941	GREED
0942	GREEDY
0943	GREEK
0365	GREEN
0944	GROUND

0945	GROUP
0946	GROW
0947	GROWTH
0949	GUERILLA
0951	GUIDE
0952	GUIDED
0953	GUIDEDMISSILE
0954	GUILTY
0956	GUNS
1507	H
0958	HABIT
0959	HAD
0960	HAIL
1422	HANG
1467	HANGUP
0963	HARBOR
0965	HARDCOPY
0968	HAS
0969	HATRED
0970	HAVE
0971	HAZE
0972	HE
1415	HEAD
0975	HEALTH
0976	HEAR
1463	HEIGHT
0981	HELP
0982	HER
0361	HIGH
0983	HIGHER
0984	HIGHWAY
0985	HILL
0986	HIM
0987	HINDER
0988	HISTORY
1405	HOLD
0989	HOLLAND
0990	HOME
0991	HOPE
0344	HORIZONTAL
0992	HOSPITAL
0993	HOT
1807	HOTEL
0994	HOUR
1607	HOW
0996	HOWEVER
0997	HUGE
0998	HUMID
0318	HUNDRED
0999	HUNGARY
1000	HURRICANE
1002	HYDROGEN
1508	I

1003	ICE
1004	ICELAND
1005	IDEA
1007	IDEOLOGY
1008	IDLE
0382	IF
1010	IMPLICATIONS
1011	IMPORTANT
0130	IN
0383	INCREMENT
0373	INDEX
1014	INDIA
1015	INDICATE
1016	INDICATED
1017	INDONESIA
1022	INFERIOR
1023	INFORM
1025	INLET
1026	INNER
1027	INNOCENT
0323	INSERT
1031	INSTANT
1034	INTEND
1035	INTENSE
1036	INTENSITY
1037	INTERACTIVE
1040	INTERNAL
1042	INTO
1043	IRANIAN
1044	IRAQ
1045	IRAQI
1046	IRELAND
1048	ISOLATED
1049	ISOLATION
1050	ISRAELI
1051	IT
1052	ITALIAN
1608	ITEM
1053	ITSELF
1509	J
1054	JAIL
1609	JAKE
1055	JANUARY
1056	JAPAN
1057	JERUSALEM
1058	JEWISH
1059	JORDAN
1060	JORDANIAN
1061	JOURNAL
1809	JULIET
1062	JULY
1064	JUNE
1065	JUST

1000 Word List

1510	K
1810	KILO
1068	KINDLY
1610	KING
1416	KNEES
1511	L
0380	LABEL
1069	LAKE
1070	LAND
1071	LANDING
1072	LARGE
1073	LARGELY
1074	LAST
1076	LATITUDE
1077	LAUNCH
1080	LEADER
1081	LEANING
1082	LEARN
1083	LEBANON
1084	LEBANESE
0331	LEFT
1086	LET
1087	LETTER
1425	LETTERS
1088	LEVEL
1089	LIBYA
1090	LIBYAN
1411	LIGHT
1091	LIGHTS
1092	LIKE
1811	LIMA
1093	LIMIT
0381	LINE
0342	LINEPLOT
1097	LIRA
1098	LISTEN
1099	LITTLE
1100	LIVE
1101	LIVING
0055	LOCAL
1457	LOGARITHM
1107	LONG
1108	LONGITUDE
1109	LOSS
1611	LOVE
0360	LOW
1110	LOWER
1512	M
1113	MACHINE
1114	MADE
0368	MAGENTA
1115	MAGNETIC
1116	MAIN

1118	MAJOR
0336	MAKE
1119	MALAYSIA
1120	MAN
1121	MANNING
1123	MANY
1124	MARCH
0386	MARGIN
1125	MARINE
1437	MARK
1127	MATTER
0358	MAXIMUM
1128	MAY
1130	ME
1131	MEAN
1132	MEANT
1135	MECHANICAL
1137	MEDICINE
1138	MEDITERRANEAN
1140	MEMORY
1141	MEN
1142	MENU
1143	MERCHANT
1144	MERIDIAN
1145	MERIT
1146	MESSAGE
1148	MEXICO
1150	MIDDLE
1612	MIKE
0320	MILLION
1152	MINE
1155	MINIMUM
1156	MINOR
1451	MINUS
1157	MINUTE
1160	MISSING
1161	MISSION
1162	MISTER
1164	MODELLED
1166	MODIFY
1167	MOMENT
1169	MONTH
1170	MOON
1171	MORE
1172	MOROCCO
1173	MOST
1177	MOUNTAIN
1179	MOVE
1180	MOVEMENT
1513	N
0362	NAME
1613	NAN
1184	NATION

1185	NATIONAL
1186	NATIONALIST
1187	NATIVE
1189	NAVAL
0327	NEGATIVE
1191	NEGOTIATE
1192	NEITHER
1194	NEVER
1195	NEW
0326	NEWLINE
0119	NEXT
1196	NIGHT
0009	NINE
0309	NINETEEN
0317	NINETY
0010	NO
1197	NOISY
1198	NOR
0051	NORTH
0056	NORTHEAST
0057	NORTHWEST
1199	NORWAY
1200	NOT
1201	NOTICE
1202	NOVEMBER
1203	NOW
1204	NUCLEAR
1205	NUMBER
1466	NUMBERS
1423	NURSE
1514	O
1614	OBOE
1209	OCEAN
1210	OCTOBER
1211	OF
1406	OFF
1212	OFFENSE
1215	OFFICE
1216	OFFICER
1217	OLD
1420	ON
1219	ONCE
0001	ONE
1220	ONLY
1221	OPEC
1222	OPEN
1224	OPERATE
1226	OPPORTUNITY
1228	OR
1229	ORDER
1231	ORDINARY
1233	ORIGINALLY
1814	OSCAR

1234	OUR
1235	OUT
1236	OUTER
1237	OVAL
1238	OVER
1240	OWN
1515	P
1243	PACIFIC
0387	PAGE
1245	PALESTINIAN
1815	PAPA
1434	PARENTHESIS
1250	PAST
0375	PAUSE
1253	PEAK
1254	PENDING
1255	PENINSULA
1256	PEOPLES
1258	PERHAPS
1429	PERIOD
1260	PERMANENT
1615	PETER
1446	PI
0332	PICA
1265	PHILLIPINES
1267	PLAINS
1268	PLANETS
1271	PLEASE
1450	PLUS
1439	POINT
1273	POLAND
1274	POLE
1275	POLITICAL
1276	POND
1277	POOL
1278	PORTUGAL
1280	POSITIVE
1283	PRECIOUS
1287	PRESENT
1288	PRESIDENTIAL
1289	PRIMARY
1291	PRIOR
1293	PROBABLY
1428	PUNCTUATION
1516	Q
1302	QUALITY
1816	QUEBEC
1616	QUEEN
1305	QUERY
1436	QUESTION
1306	QUEUE
1307	QUIET
1435	QUOTATION

1517	R
1407	RADIO
1311	RAILROAD
1312	RAIN
1313	RAISE
0339	READ
1314	REAR
1315	RECALLED
1316	RECEIVE
1317	RECEIVED
0364	RED
1321	REFERENCE
1323	REFORMAT
1325	REFUGEE
1329	RELATED
1330	RELEASE
1331	REMIND
1332	REMINDER
1334	RENAME
1336	REPEAT
1337	REPORT
1338	REROUTE
1342	RESIDENT
1343	RESIST
1344	RESPONSE
1345	RESPONSIBLE
0376	RESUME
1346	RETREAT
1347	RETRIEVAL
1348	RETRIEVE
1417	RETURN
1351	RIVER
1352	ROAD
1617	ROGER
1355	ROMANIA
1817	ROMEO
0347	ROTATE
1356	ROUGH
1357	ROUND
1358	ROUNDS
1359	RULER
1360	RUN
1361	RUNNING
1362	RUNWAY
1363	RUSSIA
1364	RUSSIAN
1518	S
1365	SAID
1618	SAIL
1367	SAME
1369	SAUDI
1371	SAY
1372	SCALE

1374	SCIENTIFIC
1375	SCOTLAND
0357	SCREEN
1376	SEA
1377	SECOND
1378	SECONDARY
1381	SEE
1384	SEEMED
1431	SEMICOLON
1386	SEND
1387	SENT
1388	SEPARATE
1389	SEPTEMBER
1473	SET
1393	SETTLEMENT
0007	SEVEN
0307	SEVENTEEN
0315	SEVENTY
1394	SHALL
1395	SHALLOW
1396	SHATTERED
1397	SHE
0389	SHIFT
1478	SHORT
1479	SHOULD
0337	SHOW
1818	SIERRA
1481	SILENT
1482	SINCE
1453	SINE
00	SIX
0306	SIXTEEN
0314	SIXTY
0377	SKIP
1485	SLEET
1486	SLOW
1487	SLOWLY
1488	SMALL
1489	SMOOTH
1490	SO
1491	SOCIALIST
1492	SOME
1493	SOON
0052	SOUTH
0058	SOUTHEAST
0059	SOUTHWEST
1494	SOVIET
1427	SPACE
1495	SPAIN
1498	SPHERICAL
0359	SQUARE
1458	SQUAREROOT
1527	STARS

1528	START
1529	STARTED
0385	STARTOVER
1410	STATION
1531	STEAM
1461	STEP
1475	STOP
1535	STORE
1536	STORM
1538	STRAIGHT
1539	STRATEGIC
1540	STREAM
1542	STUDY
1544	SUBDUE
1546	SUBTRACT
1567	SUCH
1568	SUEZ
1569	SUFFERING
1570	SUN
1571	SUNSHINE
1574	SUPPORT
1575	SURELY
1578	SUSPENDED
1580	SWEDEN
1581	SWITZERLAND
1583	SYRIA
1584	SYRIAN
1519	T
0388	TAB
1585	TABLE
[illegible]	TAIWAN
1588	TAKE
1589	TALL
1455	TANGENT
1819	TANGO
1590	TANKS
1591	TAPE
1619	TARE
1592	TARGET
1594	TECHNICAL
1418	TELEPHONE
1402	TELEVISION
1596	TEMPERATURE
0300	TEN
1599	TERMINAL
1626	TERRAIN
1627	TEST
1628	TESTING
0345	TEXT
1629	THAILAND
1630	THANK
1631	THAT
1632	THE

1633	THEIR
1634	THEM
1635	THEN
1637	THESE
1638	THEY
1639	THIN
1640	THING
1641	THINK
1642	THIRD
0303	THIRTEEN
0311	THIRTY
1643	THIS
1644	THOSE
0319	THOUSAND
0003	THREE
1645	THROW
1646	TIDE
1647	TIME
1449	TIMES
1648	TIMING
1649	TITLE
1650	TIRED
1651	TODAY
1652	TOLL
1653	TOMORROW
1654	TORNADO
0350	TRACKBALL
1659	TRAGEDY
1662	TRAIN
1664	TRANSFER
0349	TRANSLATE
1667	TRANSMIT
1671	TRAP
1672	TRAVELLER
1673	TREES
1674	TRIANGULAR
1676	TROOPS
1678	TRUCK
1680	TUNDRA
1682	TURKEY
0302	TWELVE
0310	TWENTY
1683	TWICE
0002	TWO
1424	TYPEWRITER
1684	TYPHOON
1520	U
1620	UNCLE
1685	UNDER
1462	UNDERLINE
1687	UNDERWAY
1689	UNIFORM
1690	UNITEDSTATES

1691	UNTIL
0330	UP
1692	UPPER
1693	UPON
1694	UPRIGHT
1695	USA
1696	USE
1697	USER
1698	USING
1699	USSR
1521	V
1700	VAIN
1701	VALLEY
1703	VENEZUELA
0343	VERTICAL
1704	VERY
1705	VESSEL
1706	VETERAN
1621	VICTOR
1707	VIETNAM
1708	VIRTUALLY
1709	VISION
1404	VOLUME
1710	VOTE
1522	W
1711	WAIT
1712	WALK
1713	WANT
1714	WAR
1715	WARFARE
1716	WARHEAD
1719	WARPED
1720	WAS
1721	WATCH
1722	WATER
1724	WAY
1725	WE
1726	WEAPONS
0113	WEATHER
1727	WEEK
1728	WELCOME
1729	WELFARE
1730	WELL
1731	WERE
0054	WEST
1732	WHAT
1733	WHEN
1734	WHERE
1735	WHICH
1822	WHISKEY
0370	WHITE
1736	WHOSE
1826	WHY

1000 Word List

1737	WILL
1622	WILLIAM
1738	WIND
0338	WINDOW
1739	WING
1740	WISH
1741	WITH
1742	WHO
1743	WOMAN
1745	WORK
1746	WORKING
1748	WORLD
1749	WORST
1750	WOULD
1751	WOUNDED
0340	WRITE
1523	X
1623	XRAY
1524	Y
1824	YANKEE
1752	YEAR
0366	YELLOW
0011	YES
1753	YET
1624	YOKE
1754	YOU
1755	YOUNG
1756	YOUR
1757	YUGOSLAVIA
1525	Z
0000	ZERO
1625	ZED
1825	ZULU

APPENDIX B

DAMTAD Documentation

Sample User Interaction

Mount audio tape to be digitized.
Mount magtape.

Type:

.R DAMTAD <CR>	Call the program.
*EF <CR>	Initialize the tape.
*WW <CR>	Start the audio tape. Start the magtape.
*<CR>	Any keyboard input will terminate the process.
*RW <CR>	Rewind the tape.
*RR <CR>	Read back data just written to the tape (audio and visual displays).
*<CR>	Terminate.

<CR> indicates carriage return.

```
;THE TAPE FUNCTIONS ARE INITALIZED BY 'RW',
;WHICH REWINDS; THEN SPACES FORWARD OVER THE BEGINNING
;END-OF-FILE MARK; AND SETS THE CURRENT BLOCK NUMBER TO ZERO
;
;SELECTED BY THE BAKBRD INTERPRETER, THE FOLLOWING FUNCTIONS
;PERFORM CONTINUOUSLY UNTIL THE KEYBOARD IS STRUCK:
;       THE A/D CONVERTOR(CH0) IS DUMPTED TO MAGTAPE:   WW
;       THE MAGTAPE IS DUMPED TO THE D/A CONVERTOR:     RR
;       THE CORE BUFFER IS PLAYED THRU THE D/A:         PP
;       THE LOW BYTE OF THE A/D INPUT SCALED BY THE
;       SWITCHES IS DISPLAYED FOR OFFSET CHECKING:      AD
;       THE HIGH BYTE OF A/D INPUT IS DISPLAYED:        SC
;
;THE FOLLOWING FUNCTIONS POSITION THE MAGTAPE TO (AND PLAY
;CONTINUOUSLY THE EIGHT BLOCKS PRECEDING) THE BLOCK
;SPECIFIED: A SECOND IS TAKEN TO BE 8 BLOCKS.
'#' REPRESENTS A DECIMAL NUMBER.
;       ABSOLUTE BLOCK NUMBER:                          #BA
;       ABSOLUTE SECOND:                                #SA
;       BLOCK RELATIVE TO CURRENT BLOCK:                #BR
;       SECONDS RELATIVE TO CURRENT BLOCK:              #SR
;       BECAUSE THE MAGTAPE AND D/A ARE ASYNCHRONOUS,
;       SPACES BACK TO WHAT WAS JUST HEARD FROM 'RR':   SP
;
;AN END-OF-FILE MARK CAN BE WRITTEN:                    EF
;WHICH SHOULD BE DONE ONCE AT THE BEGINNING OF THE
;TAPE AS AN INITIALIZATION FOR UTRAN COMPATIBILITY
;
;THE CURRENT BLOCK NUMBER OFTEN PRINTS
;AUTOMATICALLY OR CAN BE CALLED:                        BN
;THE PEAK AMPLITUDE IS DETERMINED ANEW
;       DURING THE FUNCTIONS--WW,RR,PP,SC--AND
;       PRINTS AUTOMATICALLY OR CAN BE CALLED:          AM
;
;UNDER ERROR CONDITIONS THE MAGTAPE STATUS REGISTER IS
;PRINTED.
```

APPENDIX C

VEDIT Documentation

Sample User Interaction

Load magtape.
Load label file to match tape.

```
.R VEDIT <CR>
*OA <CR>               Open audio file (magtape)
Digitized file =
*TA0: <CR>             Name of device
*OL <CR>
*LABEL FILE =
*LABEL 6A <CR>         Name of file
*1000TH <CR>           Set threshold for amp
*8GP <CR>              Set trigger number of frames
                       to advance after trigger to
                       center video display.
*PL <CR>               Get next word, move cursor to
                       start of word by inspection
                       and allow 60 milliseconds
                       minimum of silence.
*S <CR>                Move cursor to end of word.
*E <CR>                Mark end of word.
*100SN <CR>            Write label.  "Wordname" is
*EN wordname <CR>      name of word.  100 is name of
                       subject.  SN remains valid
                       until changed.
*PL <CR>               Get next word.
(continue to end of tape)
*CL <CR>               Close label file.
^C                     Exit
```

<CR> indicates "carriage return".

COMMANDS:

	/H	HELP - TYPE OUT ALL COMMANDS ALONG WITH A BRIEF DESCRIPTION.
	DM	DEFINE A MACRO
N	EM	EXECUTE THE MACRO N TIMES
	PL	START THE CONTINOUS PLAY OF THE TAPE
N	VO	ADJUST VOLUME, N IS A SHIFT COUNT
N	GP	SET NUMBER OF FRAMES TO PROCEED AFTER BOF IS TRIGGERED
N	TH	SET THRESHOLD FOR AMPLITUDE TRIGGER
N	CU	MOVE CURSOR N LOCATIONS, 0CU MEANS USE KNOB
	S	SET BEGINNING OF WORD
	E	SET END OF WORD
N	SN	SET SUBJECT NUMBER
N	WN	SET WORD NUMBER
	SWNAME.	ENTER WORD NAME AND SET WORDNO
	LNNAME.	CONVERTS A NAME TO A WORD NUMBER
N	WD	SET DIALECT OF THIS WORD TO N
N	SD	SET DIALECT OF ALL WORDS OF THIS SUBJECT IN LABEL FILE
N	NL	CONVERTS A WORD NUMBER TO A NAME
	OA	OPEN RAW A/D FILE
	PB	PLAY THE ENTIRE CORE BUFFER
	PC	PLAY THE BUFFER FROM THE CURSOR TO END
	PW	PLAY THE MARKED WORD
N	BN	LOCATE AT BLOCK NUMBER N AND PLAY BUFFER
	OL	OPEN THE LABEL FILE
	IL	INITIALIZE LABEL FILE, WRITE AN INIT EOF RECORD
	FL	WRITE END OF LABEL RECORD AT THIS SPOT
	RL	READ NEXT LABEL
	RP	READ NEXT LABEL BUT DON'T PLAY
N	WL	WRITE THE LABEL FOR THE CURRENT WORD, NAME IT N
	ENNAME.	ENTER (WRITE) LABEL. CALL IT "NAME"
	UL	UPDATE LABEL FILE, WRITE NEW LABEL BUT NO EOF
	AL	UPDATE LABEL FILE, CHANGE REC AFTER CURRENT REC
	BL	GO TO BEGINNING OF LABEL FILE
	EL	GO TO END OF LABEL FILE
N	MP	MOVE N LABELS IN THE LABEL FILE BUT DON'T PLAY
N	ML	MOVE N LABELS IN THE LABEL FILE AND PLAY WORD
N	LS	LOCATE SUBJECT NAMED N
N	LW	LOCATE WORD N OF CURRENT SUBJECT
N	NW	LOCATE NEXT EXAMPLE OF WORD N
	CW	COPY LABEL OF THIS WORD TO END OF SECONDARY FILE
	CL	CLOSE THE LABEL FILE
N	SC	SET THE SCALE FACTOR FOR THE FINE DISPLAY
N	DS	START THE DISPLAY
		USING NON-ZERO ARGUMENT CHANGES TO FINE DISPLAY
	OD	CALL ODT
N	KE	UNLOCK COMMANDS DM,EM,IL,FL,OD,AL (KEY IS 32000)
N	<CR>	CARRIAGE RETURN, MOVE N BLOCKS AND PLAY THE MARKED WORD
	OS	OPEN SECONDARY LABEL FILE

```
                NOTE - SECONDARY FILE WILL BE ACTIVE
        SS      SWITCH TO SECONDARY LABEL FILE
        PP      SWITCH BACK TO PRIMARY LABEL FILE
     N  SI      PLAY N/100 SECONDS OF SILENCE
     N  PD      QUICKLY ACCESS AND PLAY THE NEXT N (BUT AT LEAST
                ONE) WORDS INDICATED BY THE LABEL FILE
                DIRECTLY FROM THE RAW A/D FILE
        PR      PRINT ENTIRE LABEL RECORD ON TT
        8PR     PRINT LABEL RECORD IN OCTAL
.ENDR
.REPT 0
                LINKING INSTRUCTIONS
        .R LINK
        *VEDIT<VEDIT,ODTX,UTRT11.NOE,SY:SLIBR,DK:VEDBUF
        *C


        ** NOTE **

        VEDBUF MUST BE THE LAST FILE IN THE LINK.

        UTRT11 IS THE NO EAE VERSION. FOR INSTRUCTIONS ON GETTING
        IT, ASK ONE OF THE PROGRAMMERS.
        .R MACRO
        UTRT11.NOE<FAKEAE.PRE,F.MAC,UTRT11.MAC
        *^C
```

APPENDIX D

LBLLST Documentation

```
C             LBLLST (LABEL LIST) IS DESIGNED TO MAKE LISTINGS
C       FROM LABEL FILES CREATED BY VEDIT IN TWO BASIC WAYS:
C
C             1. IT CAN LIST ALL THE WORDS SAID BY EACH SUBJECT.
C             2. IT CAN LIST THE NUMBER OF TIMES EACH WORD WAS
C                SAID (TOKENS/TYPE). THIS CAN OPTIONALLY BE RE-
C                QUESTED FOR EITHER ALL FILES GIVEN AS INPUT
C                COLLECTIVELY OR FOR EACH INPUT FILE.
C
C       THE NAMES.COH FILE IS USED TO LOOK UP THE WORDS
C       CORRESPONDING TO THE WORD NUMBERS FOUND IN THE
C       LABEL FILES.
C
C       GENERAL PROCEDURE---------------
C
C             THE MAIN PROGRAM IS A VEHICLE FOR REQUESTING
C       INPUT AND OUTPUT FILES AND FINDING WHICH LISTINGS ARE
C       DESIRED. THE SUBROUTINE LOCATE FINDS THE POSITION OF
C       EACH WORD NUMBER IN THE NAMES FILE (AS OFFSET FROM THE
C       BEGINNING OF THE FILE) AND ENTERS IT INTO THE ARRAY
C       WRDMEM USING THE WORD NUMBER ITSELF AS THE INDEX.
C
C             SUBWRD LOCATES SUBJECT NUMBERS IN THE INPUT
C       LABEL FILES AND PUTS ALL WORD NUMBERS CITED AFTER
C       CONSECUTIVE OCCURRENCES OF A SPECIFIC SUBJECT NUMBER

C       INTO THE ARRAY NUMWRD.   WHEN A NEW SUBJECT NUMBER
C       IS COME ACROSS OR A PARTICULAR FILE ENDED (AS
C       SIGNALLED BY THE OCCURRENCE OF 8 CONSECUTIVE BLANK
C       (0) WORDS) OUTSUB IS CALLED. OUTSUB PRINTS THE
C       SUBJECT NUMBER AND ALL THE WORDS SAID BY THAT
C       SUBJECT.  THE SUBROUTINE GETWRD USES THE ARRAY
C       WRDMEM (FORMED IN LOCATE) TO RETRIEVE THE WORD
C       CORRESPONDING TO EACH WORD NUMBER STORED IN
C       NUMWRD. IF THE WORD NUMBER WAS NOT
C       FOUND IN THE NAMES FILE USED, "NO WORD NUM" IS
```

LBLLST Documentation

```
C        OUTPUTTED IN PLACE OF THE WORD.

C                TOTAL READS EACH WORD NUMBER IN THE LABEL FILES
C        AND USES IT AS THE INDEX FOR THE ARRAYS ALLWRD AND
C        LSTWRD, WHICH TOTAL ALL THE OCCURENCES OF A WORD IN
C        ALL INPUT FILES AND ALL THE OCCURRENCES OF A WORD IN

C        THE PARTICULAR FILE BEING READ, RESPECIVELY. OUTTOT
C        IS PASSED THE ARRAY LSTWRD (IF A PER FILE LISTING IS

C        REQUESTED) AFTER EACH FILE IS GONE THROUGH AND
C        OUTPUTS (USING GETWRD) THE WORD FOLLOWED BY THE
C        NUMBER OF TIMES THAT WORD OCCURRED IN THE FILE.
C        WHEN ALL FILES ARE FINISHED, ALLWRD IS PASSED
C        TO OUTTOT (IF A GRAND TOTAL COUNT IS REQUESTED).


C        NOTES: COMPILING FORMAT:
C                LBLLST=LBLLST/E/N:17
C               LINKING FORMAT:
C                LBLLST=LBLLST,ASSIGN,NOWORD,SLIBR/F
C                IF THE FORLIB (/F) USED HAS THE VESION OF ASSIGN
C                WHICH TAKES TTY INPUT, THE ASSIGN MODULE MAY
C                BE OMITTED.
C               IT IS ASSUMED THE READER HAS KNOWLEDGE OF THE
C               STRUCTURES OF THE NAMES AND LABEL FILES.
C               IF NOT, FAMILIARIZATION WITH THESE WOULD BE
C               HELPFUL IN UNDERSTANDING THE WORKINGS OF THE
C               PROGRAM THOUGH THE ESSENTIAL FEATURES ARE BRIEFLY
C               DESCRIBED IN THE APPROPRIATE PLACES BELOW.
C               THE PROGRAM MUST BE LINKED WITH NOWORD, SLIBR, AND
C               FORLIB.
```

APPENDIX E

S4I36 Documentation

DOCUMENTATION FOR S4I36 AND SPIN36

THIS IS THE DRIVING PROGRAM FOR THE SINGLE CHANNEL ISOLATED WORD ALGORITHM. IT IS USED FOR DEBUGGING AND CHECKOUT OF THE ALGORITHM, STAND-ALONE RECOGNITION AND DEMONSTRATION, ALGORITHM DEVELOPMENT

AND ENHANCEMENT AND POINT PICKING FOR REFERENCE PATTERN GENERATION.

S4I36 AND SPIN36 CAN BE ASSEMBLED WITH A VARIETY OF OPTIONS. THE CURRENT STANDARD VERSIONS ARE:

S4IC	DEBUGGING AND DEVELOPEMENT, CORRELATION FILE FUNCTIONS
S4IA	POINT PICKING AND REFERENCE PATTERN GENERATION RAW A/D FILE FUNCTIONS
S4IP	RECOGNITION ENVOLVING LARGE REFERENCE PATTERN FILES

ASSEMBLY INSTRUCTIONS FOR S4I36:

```
S4I36<S4IC,CSECTI,S4I36
S4I36<S4IA,CSECTI,S4I36
S4I36<S4IP,CSECTI,S4I36
```

ASSEMBLY INSTRUCTIONS FOR SPIN36:

```
SPIN36<S4IC,CSECTI,SPIN36
SPIN36<S4IA,CSECTI,SPIN36
SPIN36<S4IP,CSECTI,SPIN36
```

LINK INSTRUCTIONS:

```
S4IC<S4I36,VNEWL,SPIN36,SY:SLIBR
```

```
S4IA<S4I36,VNEWL,SPIN36,SY:SLIBR
S4IP<S4I36,VNEWL,SPIN36,SY:SLIBR
```

ALL VERSIONS MAY BE ASSEMBLED AND LINKED BY RUNNING THE BATCH FILE S4I36.BAT

```
;COMMANDS:

;      GO      RECOGNIZE A WORD USING ANALOG INPUT
;    N AD      TRAIN PATTERN NAME N WITH LAST WORD SPOKEN
;    N AW      TRAIN PATTERN  N WITH LAST WORD SPOKEN
;    N PR      TYPE WORD NAME  N ON CONSOLE
;    N AU      INCREMENT NC FOR NEXT COMMAND IN MACRO USING
;              COUNTER N
;      DS      START THE DISPLAY
;      CL      CLEAR OUT ALL BUFFERS
;      DL      DISPLAY LOG FRAMES ONLY(USEFUL IN NOEAE VERSION)
;      PL      PLOT THE CURRENT DISPLAY
;      FT      DRAW MOIRE PATTERN FORMAT TRACKS
;    N DB      SET OR RESET A BREAK POINT TO CALL THE DEBUGGER
;              AT DEBUG POINT N IN SPIN36
;    N SL      SCALE THE SIZE OF THE LOG FRAMES IN THE DISPLAY
;    N SA      SCALE THE SIZE OF THE AMPLITUDE CURVE DISPLAYED
;    N SS      SCALE THE SIZE OF THE SUBJECTIVE TIME CURVE
;              DISPLAYED
;    N SP      SCALE THE SIZE OF THE POWER SPECTURM FRAMES
;              DISPLAYED
;    N SH      SHIFT RAW A/D FRAME UP IN AMPLITUDE
;    N ST      SHIFT RAW A/D FRAME IN TIME
;    N FR      MOVE THE CURSOR TO FRAME N AND DISPLAY POWER
;              SPECTRUMS STARTING AT THAT FRAME
       KB      TURN KNOB ON
;    N NR      SET NUMBER OF LIKLIHOOD SCORES TO PRINT
;      KS      PRINT OUT SCORE(PERCENT RIGHT) AND START OVER
;    N NW      SET NUMBER OF WORDS IN VOCABULARY
;      WS      DEFINE THE NAMES OF THE WORDS IN THE VOCABULARY
;              (WL10,11,0,1,2,3...)
;      NM      READ IN NAME FILE WITH LIST OF WORD NAMES
;      AN      ANSWER OR HANGUP THE PHONE
;      NS      STOP COMPUTING STATISTICS AFTER RECOGINIZING A
;              WORD
;      NP      STOP PRINTING AMPLITUDE, DURATION THRESHOLD
;              INFORMATION
;    N FS      DEFINE THE N STATISTICS FILES CONTAINING THE
;              VOCABULARY
;    N TF      DEFINE N STAT FILES AND N TRAINING FILES
;      TR      SPECIFY THE FACTOR ANALYSIS TRANSFORMATION FILE
;      FF      SPECIFY A NONSTANDARD FOURIER TRANSFORM MATRIX
;              FILE
;      OL      OPEN LABEL FILE
;      BL      GO TO BEGINNING OF LABEL FILE
;    N ML      SPACE IN LABEL FILE (FORWARD OR BACKWARD)
;      CL      CLOSE LABEL FILE
```

```
;          RL       READ NEXT LABEL
;          OA       OPEN RAW A/D FILE
;          CA       CLOSE RAW A/D FILE
;          RA       RECOGNIZE FROM RAW A/D FILE USING CURRENT LABEL
;                   AS POSITION
;          GA       RECOGNIZE FROM RAW A/D FILE USING CURRENT
;                   POSITION OF TAPE
;          RB       RECOGNIZE FROM RAW A/D FILE USING CURRENT LABEL
;                   AS THE POSITION BUT DON'T CHECK FOR END OF WORD
;                   USING LENGTH FROM LABEL FILE
;          PP       RECOGNIZE A WORD USING AUTO CORR FRAMES IN CORE
;          AC       READ AUTO CORR FRAMES INTO JIN BUFFER FROM ANALOG
;                   INPUT
;          OC       OPEN THE RAW AUTO CORRELATION FILE
;          BC       GO TO THE BEGINNING OF THE AUTOCORRELATION FILE
;          EC       GO TO THE END OF THE AUTOCORRELATION FILE
;        N MC       MOVE N RECORDS (SPOKEN WORDS) IN THE CORRELATION
;                   FILE
;          RC       READ A WORDS WORTH OF CORRELATION FRAMES INTO
;                   BUFFER
;          WC       WRITE A WORD OF CORRELATION FRAMES FROM BUFFER TO
;                   FILE
;          DC       DELETE A RECORD (SPOKEN WORD) FROM THE
;                   CORRELATION FILE
;          CC       CLOSE THE CORRELATION FILE, AN END OF FILE RECORD
;                   IS WRITTEN ONLY IF THE CORRELATION FILE WAS
;                   WRITTEN INTO
;          OP       OPEN THE PICKED POINT FILE
;          BP       GO TO BEGINNING OF PICKED POINT FILE
;          EP       GO TO END OF PICKED POINT FILE
;        N MP       BACKSPACE N RECORDS (SPOKEN WORDS) IN THE PP FILE
;          RP       READ FROM PICKED POINT FILE AND RECOGNIZE THE
;                   WORD
;          WP       WRITE PICKED POINT RECORD FROM 12 FRAMES IN LOG
;                   BUFFER
;          KP       WRITE PICKED POINT RECORD,
;                   BUT FIRST ASK FOR OPERATOR CONFIRMATION
;          CP       CLOSE THE PICKED POINT FILE
;          DM       DEFINE A MACRO (COMMANDS THAT FOLLOW COMPOSE THE
;                   MACRO UNTIL A DOUBLE CARRIAGE RETURN IS
;                   ENCOUNTERED)
;        N EM       EXECUTE THE MACRO N TIMES
;          EX       EXIT THE PROGRAM AND CLOSE ANY OPEN FILES
;          OD       GO TO ODT

;FILES:

;          FILTER FILE -- A 32 X 32 BYTE PACKED FILE USE TO PERFORM
;                         THE FOURIER TRANSFORM OR OTHER
;                         SPECTRAL ANALYSIS
;
;          TRANSFORM FILE -- A 32 X 32 BYTE PACKED FILE USED TO
;                         PERFORM THE CLUSTERING
```

```
;                       TRANSFORMATION

;       STATISTICS FILE -- A FILE OF MAX SIZE = STSIZE CONTAINING
;                       REFERENCE PATTERNS
;               THE FORMAT IS AS FOLLOWS:
;               NVOCAB          NUMBER OF PATTERNS IN THIS FILE
;               NFS             NUMBER OF ELEMENTS PER PATTERN
;               NBYTE           NUMBER OF BYTES PER ELEMENT
;               WORD NAMES      SIZE = NVOCAB
;               MEANS           SIZE = NVOCAB X NFS X NBYTE
;               STANDARD DEVS   SIZE = NVOCAB X((NFS X NBYTES)+4)

;       NAME FILE --

;       1-256.  UNUSED (PUT OUT BY THE LINKER IF NAME FILE IS
;                       BUILT BY ASSEMBLING A SOURCE FILE OF
;                       SPECIFIED FORMAT)
;       1:      NUMBER OF NAME IN NAME FILE
;       2-N     NAMES OF WORDS


;       RAW AUTO CORRELATION FILE --

;       HEADER FORMAT:
;       1:      PTR TO NEXT HEADER IN FILE (WORD COUNT - STARTS
;               AT 1)
;       2:      POINTERS ARE DOUBLE PRECISION
;       3:      IARG, NO OF FRAMES (DATA RECORDS) IN WORD
;       4:      36., LENGTH OF FRAME
;       5:      2, POINTERS ARE DOUBLE PRECISION
;       6:      SUBJECT NO
;       7:      WORD NO
;      33:      PTR TO LAST HEADER RECORD
;      34:      POINTERS ARE DOUBLE PRECISION
;      35:      PTR TO CURRENT HEADER RECORD
;      36:      POINTERS ARE DOUBLE PRECISION

;       DATA RECORD FORMAT:
;       1:      SHIFT COUNT
;       2:      AUTO CORRELATION TERMS
;       34:     AMPLITUDE
;       35:     AMPLITUDE
;       36:     AMPLITUDE


;       PICKED POINT FILE -- CONTAINS 12 LOGTRANSFORMED FRAMES
;                       FOR EACH WORD IN THE DATA BASE

;       HEADER FORMAT:
;       1:      PTR TO NEXT HEADER IN FILE (WORD COUNT - STARTS
;               AT 1)
```

```
;
;         2:       POINTERS ARE DOUBLE PRECISION
;         3:       12, NUMBER OF PICKED POINTS PER WORD
;         4:       36., LENGTH OF FRAME
;         5:       2, POINTERS ARE DOUBLE PRECISION
;         6:       SUBJECT NO
;         7:       WORD NO
;        33:       PTR TO LAST HEADER RECORD
;        34:       POINTERS ARE DOUBLE PRECISION
;        35:       PTR TO CURRENT HEADER RECORD
;        36:       POINTERS ARE DOUBLE PRECISION

;         DATA RECORD FORMAT:
;         1:
;         2:
;         3:       SUBJECTIVE TIME
;         4:       REAL TIME
;         5:       LOG TRANSFORMED FRAME


;         LABEL FILE --

;         1:       SUBJECT NUMBER
;         2:       WORD NAME NUMBER
;         3:       POINTER TO LOCATION IN RAW A/D FILE OF BEGINNING
;                  OF WORD
;         4:       POINTERS ARE DOUBLE PRECISION
;         5:       LENGTH OF WORD IN FRAMES (79. WORDS = 1 FRAME)
;         6:       UNUSED
;         7:       UNUSED
;         8:       UNUSED
```

STANDARD INTERACTION:

```
          < > SPECIFIES OPTIONAL COMMAND

          .R S4IC
          <*FF>
          <FILTER FILE=>
          <*[ENTER COSINE MATRIX FILE]>
          *DS               START DISPLAY

          *FS
          STAT FILE=
```

```
        *[ENTER STAT FILE NAME]

        *OC
        CORR FILE=
        *[ENTER RAW CORRELATIONS FILE]

---

        *RC                 READ ONE WORD FROM CORRELATION FILE
        *PP                 RECOGNIZE IT

        <*BC>               GO TO BEGINNING OF CORRELATION FILE
        <*'N'MC>            SPACE N WORDS IN CORRELATION FILE

        OR

        *GO                 RECOGNIZE LIVE INPUT

---

        .R S4IA

        <*FF>
        <FILTER FILE=>
        <*[ENTER COSINE MATRIX]>

        *OP
        PICKED POINT FILE=
        *[ENTER PICKED POINT FILE NAME]

        *OA
        RAW A/D FILE=
        *[ENTER RAW A/D FILE]

        *OL
        LABEL FILE=
        *[ENTER LABEL FILE]

        *NM
        NAME FILE=
        [ENTER NAME FILE]

        *DMRLRAWO           READ LABEL, RECOGNIZE LABELED WORD,
                            SAVE PICKED POINTS
        *

        *RA                 RECOGNIZE FISRT WORD
        *WP                 WRITE FIRST WORD'S PICKED POINTS
        *DS                 START DISPLAY
        *NP                 DON'T PRINT AMPLITUDE DATA
```

```
*10000EM
*CP                 CLOSE PICKED POINT FILE

.R S4IP

<*FF>
<FILTER FILE=>
<*[ENTER FILTER FILE NAME]>

*OP
*FS
STAT FILE=
*{ENTER STAT FILE NAME]
```

```
<*FF>
<FILTER FILE=>
<*[ENTER COSINE MATRIX FILE NAME]>

*GO                 RECOGNIZE LIVE INPUT

OR

*OP
PICKED POINT FILE=
*{ENTER PICKED POINT FILE NAME]

*DMMPRPO            PRINT OUT PICKED POINT NAME AND RECOGNIZE
*                   THE WORD

<*NP>               NO PRINT
<*NR>               NO STAT RANKING

*10000EM
*KS                 PRINT OUT   RIGHT
```

Power Spectrum Calculation

Every 10 milliseconds the hardware produces an autocorrelation frame. Each autocorrelation frame undergoes processing, the result of which is a power spectrum frame, a graph of frequency vs. amplitude which which tells us what frequencies were present in the 10 millisecond frame.

Every 10 milliseconds a 32 term autocorrelation frame is read from the hardware "autocorrelator". Each term is 32 bits long.

The frame is smoothed in time with its two neighbors ("Triangular Smoothing"). This eliminates transient noise and pitch aliasing.

The "second derivative" of the frame is taken to maintain precision at high frequencies.

The 32 values in the frame are scanned. The maximum value is called the "Power".

The square root of the power is taken. It is called the "Amplitude".

The frame is "power normalized". Each term is divided by the power. The resulting 16 most significant bits of each term are saved. This reduces the amount of computation to be done and data stored.

The 32 term frame is multiplied by a 32 by 32 "cosine matrix". This is called a cosine transform, the result of which is a 32 term power spectrum. The 32nd term is thrown away and replaced with the amplitude term. This 32 element frame is called th "Power Spectrum".

Triangular Smoothing

The 3 time-adjacent frames are averaged. For each lag the smoothed value is equal to the first frame plus two times the second frame, plus the third frame.

The triangular method replaces the previous 5 frame moving average which smoothes five neighboring frames (each

given equal weight).

The 3 frame method was chosen because it does not smooth away fast transitions and better maintains independence between adjacent sounds.

The triangular method also provides some protection against aliasing produced by the 100 Hz frame rate beating against the pitch period.

Change to the triangular smoothing produced no improvement in recognition.

Autocorrelation Function

Let f be a function of time.

$$R(t) = \int_{-\infty}^{+\infty} f(x) \cdot f(x+t)\,dx \qquad \text{(E1.1)}$$

The autocorrelation function resembles the time function in shape. The most important difference is that the phase has been "thrown away". The autocorrelation function of two functions that differ only in phase will be identical.

Autocorrelator

The Dialog hardware autocorrelator or "correlator"

An analog signal is sampled every 125 micro seconds by an A/D converter. These digitized samples are fed to the autocorrelator. Every 10 milliseconds a 32 frame autocorrelation frame is output by the correlator. This frame is an estimation of the autocorrelation function of the input signal. The autocorrelation frame is defined as follows:

$$R_t = \sum_{i=1}^{79} X_i X_{i+t} \quad t = 0,1,\ldots 31 \qquad \text{(E1.2)}$$

R_0 is called the zero delay term, R_1, the first delay term, etc. The inputs to the autocorrelator are 12 bit numbers, the outputs are 32 bit numbers.

NOTE: The autocorrelator has reduced the number of terms we deal with while increasing their significance. An autocorrelation frame or correlation is the output of the autocorrelator, one 32 term frame.

Second Derivative

The purpose of the second is to maintain the high frequency resolution of the power spectrum. The result of taking the second derivative of an autocorrelation frame is the same as having applied a 6 decibel per octave filter to the input signal.

The formula for the first derivative is:

$$d_i = X_i - X_{i+1} \qquad \text{(E1.3)}$$

X_i is the ith term of th autocorrelation frame

for the second derivative it is:

$$S_i = d_i - d_{i+1}$$

$$= (x_i - x_{i+1}) - (x_{i+1} - x_{i+2}) \quad \text{(E1.4)}$$

$$= x_i - 2 x_{i+1} + x_{i+2}$$

S_{31} is defined as = S_{30}

The effect of the second derivative is based on the formula

sin'(2x) = 2cos(2x)

hence

sin"(2x) = -4sin(2x)

while

sin"(x) = -sin(x)

The effect of taking the second derivative shows up during the cosine trans where multiplying by large numbers instead of small numbers produces a more significant answer.

Amplitude

By the definition of the autocorrelation function

$$R(0) = \int_{-\infty}^{+\infty} f(x) \quad f(x)\, dx \quad \text{(E1.5)}$$

hence $\sqrt{R(0)}$ is the RMS amplitude of the f(x) and if R exists R(0) >= R(x). However the "autocorrelation frame" is the short-term estimation of the autocorrelation function R and it is not true that

$R_0 >= R_i$

(for a Ramp, R_{31} is the largest term in the frame).

To overcome this problem the amplitude is defined to be the square root of the largest term in the autocorrelation frame.

Power

The power is the square of amplitude or the maximum value in the autocorrelator.

APPENDIX F

STAT9 Documentation

```
C                         SLM, 30 OCT 72
C
C       UNBIASED, USING NO. OF DEGREES OF FREEDOM =
C       NO. OF SUBJECTS
C
C       COMPUTES ISOLATED WORD (ONE BYTE) OR CONTINUOUS SPEECH
C       (TWO BYTE) REFERENCE PATTERNS USING "PICKED POINTS"
C       DATA BASE PRODUCED BY S4I OR DBASE SERIES PROGRAMS
C
C       THE FORMATS OF THE PICKED POINT FILE AND THE REFERENCE
C       PATTERN FILE (OUTPUT FILE) ARE FOUND IN S4I36.DOC
C
C       AN INTERMEDIATE SCRATCH FILE USED AS ACCUMULATOR
C       STORAGE MUST BE PROVIDED .  THE LENGTH OF THIS
C       FILE (CALLED THE "STATMP" FILE) IS
C
C       (4*NFS+1)*NWORD
C
C       16-BIT WORDS, WHERE NFS IS THE TOTAL NUMBER OF DATA
C       ELEMENTS PER PATTERN AND NWORD IS THE TOTAL NUMBER OF
C       REFERENCE PATTERNS TO BE COMPUTED.
C
C

C       NORMAL INTERACTION FOR MAKING ISOLATED WORD PATTERNS

C       .R STAT9
C       PICKED POINT FILE=
C       *[PICKED POINT FILE]
C       STATMP FILE=
C       *[STATMP FILE]
C       OUTPUT FILE=
C       *[NEW PATTERN FILE]

C       *NM            YOU MUST SPECIFY THE NUMBER OF NAMES
C                      AND THE NAMES BEFORE TYPING "GO".
C                      THIS CAN BE DONE EITHER WITH THE "NM"
C                      OPTION OR WITH THE "NW","WS" COMBINATION.
C       NAME FILE=
```

```
C       *[NAME FILE]
C       *GO      (START PROGRAM)
C       ZERO OLD STATMP FILE?
C       *1       (YES)
C       KEEP IT AFTER USE?
C       *0       (NO)                NO MEANS DON'T CLOSE THE FILE.
C                                    1 MEANS CLOSE THE FILE.
C                                    THIS HAS NO EFFECT UNLESS THE
C                                    FILE WAS CREATED WITH THE /C:0
C                                    OR /C:-1 OPTION.
C       (MUCH CALCULATION)
C       ANOTHER PICKED POINT FILE?
C       *0       (NO)

C       A PRINTOUT FOLLOWS CONTAINING NUMBER OF PICKED
C       POINTS PER CLASS, MIN AND MAX MEANS AND STDEV
C       TERMS AND THE RANGE OF THE LOG TERMS.
C       THIS IS FOLLOWED BY A PRINT OUT OF THE NUMBER OF TIMES
C       EACH WORD APPEARED IN THE PICKED POINT FILE WHICH WENT
C       INTO MAKING UP THE REFERENCE PATTERNS

C               COMMAND MENU FOR STAT9:
C
C       N BY    SET  OF BYTES PER REFERENCE PATTERN DATA ELEMENT
C               TO N (1 OR 2) 1 IS DEFAULT.
C               1 IS FOR DISCRETE WORDS
C               2 IS FOR CONTINUOUS SPEECH

C       N NW    ;SET  OF PATTERNS TO N
C         WS    SET UP TO ENTER ID NUMBERS OF ALL PATTERNS IN THE
C               ORDER THEY WILL APPEAR IN THE OUTPUT FILE.  ENTER
C               EXACTLY "NW" NUMBERS AFTER THIS COMMAND,
C               SEPARATED BY ANY NON-NUMERIC DELIMITER.
C         NM    READ IN A FILE CONTAINING PATTERN NAMES
C               FORMAT IS:
C                       256 WORDS GARBAGE (OUTPUT BY THE LINKER)
C                       IVOCAB = NO OF PATTERNS IN VOCABULARY
C                       IVOCAB NO OF PATTERN NAMES
C               THIS FORMAT WAS CHOSEN SO THE NAME FILE CAN BE
C               BUILT WITH THE EDITOR. AN EXAMPLE:
C                       .WORD   IVOCAB
C                       .WORD   NAME1,NAME2,...NAMEN
C                       .END
C               THE EDITOR FILE IS THEN ASSEMBLED AND LINKED.
C               THE SAV IMAGE IS READABLE BY STAT9.
C         SO    SPECIAL FUNCTION: BUILD REFERENCE PATTERN FILE
C               FROM EXISTING DATA IN INTERMEDATE "STATMP" FILE
C       N SL    SUBTRACT CONSTANT N*1000 FROM LOG TERM
C       N SC    SET SPECIAL SCALE FACTOR FOR INVERTED STANDARD
C               DEVIATIONS IN REFERENCE PATTERNS
C       N MS    MASK OUT THE N-TH DATUM OF EACH 32-ELEMENT FRAME
C               AND GIVE IT A WEIGHT OF ZERO IN THE REFERENCE
C               FILE
```

```
C        N TD    SET TOTAL NUMBER OF DIMENSIONS PER PATTERN (READ
C                FROM THE BEGINNING OF THE PATTERN)
C          GO    PROCEED TO COMPUTE REFERENCE PATTERNS
C
C
```

APPENDIX G

S4IT Documentation

S4IT.DOC MR 25-Jun-78

DOCUMENTATION ON SYNTAX TREE VERSION OF S4I

COMMAND SUMMARY:
(FOR GREATER EXPLANTION AND VARIATIONS, SEE FOLLOWING PAGES).

OT	OPEN TREE FILE
1OT	OPEN AND INITIALIZE TREE FILE
CT	CLOSE TREE FILE
2JT	JUMP TO 2ND NODE
MT	MAKE TRANSITION ON BASIS OF LAST RECOGNIZED WORD
2BT	TAKE BACK TWO TRANSITIONS
XT	EXECUTE NODE SUBROUTINE
1,2IT RED	INSERT 1: RED=>2
1,2DT RED	DELETE 1: RED=>2
2PT	PRINT OUT INFO ON 2ND NODE
-PT	PRINT OUT INFO ON WHOLE TREE

IN THE S4IT VERSION OF S4I, THE USER CAN DEFINE AND USE A SYNTAX TREE TO LIMIT THE SET OF WORDS TO BE RECOGNIZED

AT ANY GIVEN POINT. EACH SYNTAX TREE CONSISTS OF A SET OF NUMBERED NODES AND NAMED BRANCHES BETWEEN THE NODES. THE NAME OF A BRANCH CORRESPONDS TO A VOCABULARY WORD THAT CAN CAUSE A TRANSITION BETWEEN THE SPANNED NODES. THE NUMBER OF NODES, NUMBER OF BRANCHES, AND THEIR LINKING IS ARBITRARY (A CORE BUFFER (BRBUF) MUST BE ALLOCATED AS LARGE AS THE MAX. NUMBER OF BRANCHES PER NODE. ITS DEFAULT SIZE IS 256.). ONCE A TREE HAS BEEN GENERATED (ACTUALLY AN ARBITRARY GRAPH), A STARTING NODE CAN BE SPECIFIED AND A PATH TAKEN FROM THE STARTING NODE CAN SAVED (IN CASE YOU WANT TO BACK UP).

RUN-TIME COMMANDS EXIST FOR INVOKING A TREE, MOVING AROUND IN A TREE, CREATING AND EDITING A TREE, PRINTING OUT INFO ON A TREE, AND EXECUTING SUBROUTINES ASSOCIATED WITH NODES IN A TREE.

EXPLANATION OF COMMANDS:

NOTE THAT ALL NODES MUST BE POSITIVE. "-" (OR A NEG. NUMBER) USED AS A NODE WILL MATCH ALL NODES. IN THE EXAMPLES BELOW, "*" AND "$" ARE PROMPTS FOR TTY.

(1) INVOKING A TREE:

```
*OT                     OPEN TREE FILE
TREE FILE=
*<ENTER TREE FILE>
NAME FILE=
*<ENTER NAMES FILE>
```

IF THIS COMMAND IS PRECEEDED BY A "1", IT INITIALIZES THE TREE FILE (DONE WHEN CREATING A NEW TREE).

```
*CT                     CLOSE TREE FILE
```

THIS OR EX (EXIT) MUST BE DONE IF ANY CHANGES TO THE TREE ARE SURE TO BE SAVED IN THE TREE FILE.

(2) MOVING AROUND IN A TREE:

(A) DEFINING CURRENT STARTING NODE:

```
*3JT                JUMP TO 3RD NODE
```

(B) SPEECH-DRIVEN TRANSITIONS:

```
*2MT                MAKE TRANSITION ON BASIS OF 2ND
                    MOST LIKELY RECOGNIZED BRANCH
                    (DEFAULT IS 1).
```

(C) UN-DOING TRANSITIONS:

2BT	TAKE BACK TWO TRANSITIONS (DEFAULT IS 1).

(3) CREATING AND EDITING A TREE:

(A) INSERTING BRANCHES:

*3,5IT $RED	INSERTS BRANCH "RED" FROM NODE 3 TO NODE 5. (DEFAULT ARE CURRENT PARENT AND SON NODES).

(B) DELETING BRANCHES:

*3,5DT $RED	DELETES BRANCH "RED" FROM NODE 3 TO NODE 5. (DEFAULT ARE CURRENT PARENT AND SON NODES).

THE "-" ARGUMENT CAN BE USED TO MAKE GLOBAL INSERTIONS/DELETIONS, E.G.:

*-,4IT $HELP	INSERTS THE BRANCH "HELP" FROM EVERY NODE TO NODE 4.

THIS WILL NOT AFFECT TERMINAL NODES OR ANY NODES DEFINED AFTER THIS COMMAND WAS EXECUTED.

IF ANY TREE EDITING IS PLANNED, IT IS SUGGESTED THAT YOU WORK ON A COPY, RATHER THAN THE ORIGINAL, IN CASE SOMETHING GOES WRONG (E.G. SYSTEM CRASH, ERRORS, ETC).

(4) PRINTING OUT INFO ON A TREE:

*3PT	PRINTS OUT INFO ON 3RD NODE (DEFAULT IS CURRENT NODE).

3: RED=>5 BLUE=>6

"-" GIVEN AS AN ARGUMENT PRINTS OUT THE WHOLE TREE.

(5) NODE SUBROUTINES:

(A) EXECUTING NODE SUBROUTINES:

XT	EXECUTES SUBROUTINE ASSOCIATED WITH CURRENT NODE. THE DEFAULT SUBROUTINE (I.E. IF NOT DEFINED AS BELOW) IS A NOP.

(B) ASSOCIATING SUBROUTINES WITH NODES:

A SUBROUTINE MAY BE ASSOCIATED WITH A NODE BY LINKING THE SUBROUTINE IN WITH S4IT AND ENTERING

THE SUBROUTINE INTO THE "NODE TABLE." THE LATTER IS DONE BY INSERTING A MACRO CALL OF THE FORM:

```
ASSOC   NODE,SUBR
```

INTO THE ROUTINE THAT LOADS THE NODE TABLE. THE NEXT PAGE SHOWS AN EXAMPLE OF A NODE SUBROUTINE FILE THAT CONTAINS A SUBROUTINE THAT RINGS THE TTY BELL AND THE NODE TABLE LOADING SUBROUTINE. A USER DEFINED FILE OF THIS FORM WITH HIS OWN SUBROUTINES MAY EASILY BE LINKED WITH ALL THE OTHER NECESSARY FILES BY CALLING THE USER'S FILE DK:NSUBRS.OBJ, MOUNTING FLOPPY M.RILEY2, AND EXECUTING THE INDIRECT FILE FD:S4IT.COM. THIS WILL PUT S4IT.SAV, WITH THE USER'S NODE SUBROUTINES, ON DK. THE DEFAULT NODE SUBROUTINES AND THEIR LINKING ARE:

(1) 100. = BACK UP TWO TRANSITIONS.
(2) 101. = BACK UP ONE TRANSITION AND DO A "PT".
(3) 102. = BACK UP ONE TRANSITION AND RING TTY BELL.

THUS, FOR EXAMPLE, IF EVERY NODE HAS A BRANCH CALLED "NO" FROM IT TO NODE 100, THEN A SPOKEN "NO" CAN CAUSE THE TRANSITION BEFORE THE "NO" WAS SAID TO BE TAKEN BACK.

A TYPICAL MACRO MIGHT BE:

```
PTGOMTXT,
```

WHICH PRINTS INPUT CHOICES (PT), LISTENS FOR INPUT (GO), MAKES TRANSITION ON BASIS OF INPUT (MT), AND EXECUTES NEW NODE'S SUBROUTINE (XT).

ASSEMBLY INSTRUCTIONS:

```
S4I36<S4IT,CSECTI,S4I36
SPIN36<S4IT,CSECTI,SPIN36
STSWAP<S4IT,CSECTI,FBLOCK.PRE,STSWAP
MKTREE<S4IT,CSECTI,MKTREE
RNTREE<S4IT,CSECTI,RNTREE
NSUBRS<NSUBRS
```

LINKING INSTRUCTIONS:

```
S4IT<S4I36,SPIN36,STSWAP,MKTREE,RNTREE,VNEWL/C
NSUBRS,SLIBR
```

(SLIBR MUST CONTAIN "ITRAN" AND "GETLIN".)

TREE FILE FORMAT:

```
        1-2:     DOUBLE PREC. END-OF-FILE WORD COUNT

        <N-1 NODES>

        X:                N = NODE
        X+1:              K = BRANCH COUNT
        X+2-
         X+K+1:           BRANCH NAMES
        X+K+2-
         X+2K+1:          SON NODES
```

NODE SUBROUTINE FILE EXAMPLE:

```
        .TITLE   NSUBRS.MAC

; THIS FILE, TO BE LINKED IN WITH S4IT, CONTAINS THE NODE
; TABLE LOADER, THE ALLOC. OF THE NODE TABLE, A SUBR. THAT
; RINGS THE TTY BELL, AND THE LINKING OF THAT SUBR. TO NODE
; 100.
        .GLOBL   LOADND,NDBUF,NDSIZ

        .MCALL   .TTYOUT

; ASSOCIATES NODE WITH PARTICULAR SUBROUTINE

        .MACRO   ASSOC   NODE,SUBR
        MOV       NODE,R0
        ASL      R0
        MOV       SUBR,NDBUF-2(R0)
        .ENDM

;       ROUTINE THAT LOADS NODE TABLE.

LOADND: ASSOC    100.,BELL        ;RING TTY BELL
        RTS      PC


BELL:   MOV       7,R0
        .TTYOUT
        RTS      PC

NDBUF:  .BLKW    200.             ;NODE TABLE
NDSIZ:  .WORD    200.             ;NODE TABLE SIZE

        .END
```

APPENDIX H

List of Audio Tapes

Audio Tape	Tape Counter	Materials Copied	Word List	Date Copied
282	0-243	Cassette 43 S-1	ROME 2 (1000)	6/9/78
	250-623	" 43 S-2	" " "	
	230	" 44 S-2	" " "	
283	0-252	Cassette 47 S-1	ROME 2 (1000)	6/13/78
	260-647	" 47 S-2	" " "	
	0-250	" 48 S-1	" " "	
	260-648	" 48 S-2	" " "	
286	0-243	Cassette 35 S-1	ROME REQ.	
	245-591	" 35 S-2	ROME 2(NOUNS)	
	0-245	" 36 S-1	ROME REQ.	
	250-605	" 36 S-2	ROME 2(NOUNS)	
287	0-245	Cassette 39 S-1	ROME 2 (1000)	6/7/78
	250-626	" 39 S-2	" " "	
	0-244	" 40 S-1	" " "	
	250-627	" 40 S-2	" " "	
288	0-245	Cassette 41 S-1	ROME 2 (1000)	6/8/78
	250-625	" 41 S-2	" " "	
	0-243	" 42 S-1	" " "	
	250-626	" 42 S-2	" " "	
289	0-200	Cassette 37 S-1	ROME 2 (NOUNS)	
	0-248	" 38 S-1	" " "	
	250-629	" 38 S-2	ROME 2 (1000)	
292	0-245	Cassette 51 S-1	ROME 2 (1000)	6/13/78
	250-626	" 51 S-2	" " "	
	0-243	" 52 S-1	" " "	
	250-624	" 52 S-2	" " "	

293	0-248	Cassette 49 S-1	ROME 2 (1000)	6/13/78
	250-626	" 49 S-2	" " "	
	0-249	" 50 S-1	" " "	
	255-640	" 50 S-2	" " "	
295	0-243	Cassette 45 S-1	ROME 2 (1000)	6/12/78
	250-628	" 45 S-2	" " "	
	0-243	" 46 S-1	" " "	
	250-505	" 46 S-2	" " "	
297	0-247	Cassette 53 S-1	ROME 2 (1000)	6/13/78
	250-633	" 53 S-2	" " "	
	0-247	" 54 S-1	" " "	
	250-635	" 54 S-2	" " "	
316	0-245	Cassette 55 S-1	ROME 2 (1000)	6/14/78
	250-627	" 55 S-2	" " "	
	0-248	" 56 S-1	" " "	
	250-633	" 56 S-2	" " "	
322	0-246	Cassette 73 S-1	ROME 2 (1000)	6/16/78
	250-632	" 73 S-2	" " "	
	0-246	" 74 S-1	" " "	
	250-632	" 74 S-2	" " "	
323	0-243	Cassette 71 S-1	ROME 2 (1000)	6/16/78
	250-588	" 71 S-2	" " "	
	0-245	" 72 S-1	" " "	
	250-594	" 72 S-2	" " "	
324	0-243	Cassette 67 S-1	ROME 2 (1000)	6/16/78
	250-626	" 67 S-2	" " "	
	0-243	" 68 S-1	" " "	
	250-627	" 68 S-2	" " "	
325	0-249	Cassette 65 S-A	ROME 2 (1000)	6/16/78
	255-577	" 65 S-B	" " "	
	0-243	" 66 S-1	" " "	
	250-627	" 66 S-2	" " "	
326	0-246	Cassette 57 S-1	ROME 2 (1000)	6/14/78
	250-634	" 57 S-2	" " "	
	0-238	" 58 S-1	" " "	
	245-624	" 58 S-2	" " "	
327	0-243	Cassette 61 S-1	ROME 2 (1000)	6/15/78
	250-624	" 61 S-2	" " "	
	0-244	" 62 S-1	" " "	
	250-627	" 62 S-2	" " "	

328	0-247	Cassette	59	S-1	ROME	2	(1000)	6/14/78
	250-632	"	59	S-2	"	"	"	
	0-243	"	60	S-1	"	"	"	
	250-622	"	60	S-2	"	"	"	
329	0-243	Cassette	63	S-1	ROME	2	(1000)	6/15/78
	250-527	"	63	S-2	"	"	"	
	0-246	"	64	S-A	"	"	"	
	250-632	"	64	S-B	"	"	"	
333	246	Cassette	77	S-1	ROME	2	(1000)	6/20/78
	250-628	"	77	S-2	"	"	"	
	0-246	"	80	S-1	"	"	"	
334	0-249	Cassette	75	S-1	ROME	2	(1000)	6/19/78
	260-651	"	75	S-2	"	"	"	
	0-248	"	76	S-1	"	"	"	
	260-581	"	76	S-2	"	"	"	
337	0-250	Cassette	85	S-A	ROME	2	(1000)	6/21/78
	260-311	"	85	S-B	"	"	"	
338	0-250	Cassette	83	S-A	ROME	2	(1000)	6/21/78
	260-656	"	83	S-B	"	"	"	
	0-246	"	84	S-A	"	"	"	
	250-626	"	84	S-B	"	"	"	
339	0-243	Cassette	81	S-1	ROME	2	(1000)	6/21/78
	260-621	"	81	S-2	"	"	"	
	0-246	"	82	S-1	"	"	"	
	250-628	"	82	S-2	"	"	"	
347	0-667	Tape	130	S-1	ROME	2	(1000)	6/23/78
	0-12.5	"	130	S-1	"	"	"	
	20-667	"	130	S-2	"	"	"	
348	0-661	Tape	140	S-1	ROME	2	(1400)	6/21/78
	0-645	"	140	S-2	"	"	"	
349	0-654	Tape	133	S-1	ROME	2	(1400)	6/21/78
	0-661	"	133	S-2	"	"	"	
350	0-657	Tape	141	S-1	ROME	2	(1400)	6/22/78
	0-657	"	141	S-2	"	"	"	
351	0-28	Tape	117	S-2	ROME	2	(1400)	6/23/78
	30-64	"	130	S-2	"	"	"	
	70-150	"	183	S-1	"	"	"	
	160-241	"	183	S-2	"	"	"	

352	0-664	Tape	117 S-1	ROME 2 (1400)	6/23/78
	0-15	"	117 S-1	" " "	
	20-680	"	117 S-2	" " "	

APPENDIX I

100 Required Words

zero
one
two
three
four
five
six
seven
eight
nine
ten
eleven
twelve
thirteen
fourteen
fifteen
sixteen
seventeen
eighteen
nineteen
twenty
thirty
forty
fifty
sixty
seventy
eighty
ninety
hundred
thousand
million
erase
cancel
insert

forward
backward
newline
negative
cursor
down
up
left
pica
elite
format
graphics
make
show
window
read
write
point
lineplot
vertical
horizontal
text
color
rotate
scale
translate
trackball
enter
done
x
y
equals
clear
screen
maximum

square
low
high
name
black
red
green
yellow
blue
magenta
cyan
white
condition
action
index
set
limit
pause
resume
skip
endtape
call
label
line
if
increment
add
startover
margin
page
tab
shift

MISSION
of
Rome Air Development Center

RADC plans and conducts research, exploratory and advanced development programs in command, control, and communications (C^3) activities, and in the C^3 areas of information sciences and intelligence. The principal technical mission areas are communications, electromagnetic guidance and control, surveillance of ground and aerospace objects, intelligence data collection and handling, information system technology, ionospheric propagation, solid state sciences, microwave physics and electronic reliability, maintainability and compatibility.